CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Development of Fast Methods for Evaluating the Boltzmann Collision Operator Based on Discontinuous Galerkin Discretizations in the Velocity Variable, Convolution Formulation, and Fast Fourier Transform

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Mathematics

By

Jeffrey Limbacher

August 2018

The thesis of Jeffrey Limbacher is approved:

Ali Zakeri , Ph.D.	Date
Vladislav Panferov , Ph.D.	Date
Alexander Alekseenko , Ph.D., Chair	Date

California State University, Northridge

Acknowledgements

tbd

Table of Contents

List of Tables

List of Figures

ABSTRACT

Development of Fast Methods for Evaluating the Boltzmann Collision Operator Based on Discontinuous Galerkin Discretizations in the Velocity Variable, Convolution Formulation, and Fast Fourier Transform

By

Jeffrey Limbacher

Master of Science in Applied Mathematics

tbd

Chapter 1

Introduction

This thesis concerns itself with the evolution of gas flows in low density regimes. This is of particular interest to the engineering community. One example of scuh a regime is gas flows around high-altitude high-velocity objects flying through the upper atmosphere such as spacecraft and airplanes. Under such regimes, the particles impart a large amount of kinetic energy on the object causing a large transfer of heat to the object. Preventing damage to the object under these circumstances is essential. It is often difficult to replicate these conditions within a laboratory setting. This means that there is hope in the development of high fidelity solvers that can simulate the high speed gas flows around these objects to help predict the correct heating patterns. In these gas regimes, the fluid mechanical laws of Navier-Stokes and Fourier break down. In contrast, kinetic theory provide an accurate description by describing particles at the microscopic level.

Kinetic theory describes the non-equilibrium dynamics of a gas or any system comprised of a large number of particles. Kinetic equations have found applications in wide ranging applications such as rarefied gas dynamics [11] [10], radiative transfer, and semiconductors modeling. The Bolzmann equation is a kinetic equation that describes gases at the molecular level at regimes where Navier-Stokes and Fourier methods fail. Analytic solutions to the Boltzmann equation have been constructed for simple geometries and special molecular potentials. However, the complexity of the equation, along with the the complexities of boundary conditions and gas-to-gas interactions that occur in engineering and physics applications, suggest that only numerical solutions are possible. However, the complexity of the equation provides a challenge in directly computing the Boltzmann equation. It is composed of a five-

fold integral which must be evaluated at all points in space and velocity resulting in $\mathcal{O}(n^{11})$ computational cost where n is the number of discretization points in space and velocity. This thesis primarily concerns with evaluating this integral in velocity space which results in a $\mathcal{O}(n^8)$ cost. However, this is still computationally prohibitive.

In this thesis, we explore how to speed up evaluation of the collision operator within the Boltzmann equation by using a Discontinuous-Galerkin method based on the work of [12, 1, 2, 7]. This results in a computational complexity of $\mathcal{O}(n^8)$ with a $\mathcal{O}(n^5)$ storage requirement. In [3], the translational invariance was used to introduce a bilinear convolution form of the Galerkin projection of the collision operator. In the case of uniform meshes, this convolution allows us to re-write the collision operator as a convolution of multidimensional sequences. This thesis introduces a new method using the convolution theorem and Fast Fourier Transform to reduce the computational complexity of the convolution to $\mathcal{O}(n^6)$ complexity.

Chapter 2

The Boltzmann Equation

The kinematic theory of gases treats gases as composed of a large number of individual molecules that, for large periods of time, flow unimpeded. As these particles move freely through space, they collide with each other. Collisions of these particles are what drive the evolution of the gas towards equilibrium.

We consider a gas enclosed in a volume. A single molecule of this gas can be described with having position $\vec{x}$ and velocity $\vec{v}$ at a time t. For a particular time, we can describe a molecule as being within at a single point, $(\vec{x}, \vec{v})$, in 6-dmensional space known as phase space. We define the distribution function of the gas as $f(t, \vec{x}, \vec{v}) d\vec{x}\, d\vec{v}$ gives the number of particles within the range of $\vec{x} + d\vec{x}$ with velocities $\vec{v} + d\vec{v}$.

In 1872 Boltzmann [5] introduced the Boltzmann equation which describes the time evolution of the distribution f. In the absence of external forces and collisions of particles, then the Boltzmann equation takes the form

$$\frac{\partial}{\partial t} f(t, \vec{x}, \vec{v}) + \vec{v} \cdot \nabla_x f(t, \vec{x}, \vec{v}) = 0\,. \tag{2.1}$$

However, when the effects of collisions cannot be neglected, the right hand side must be modified. In this case, the Boltzmann equation takes the form of

$$\frac{\partial}{\partial t} f(t, \vec{x}, \vec{v}) + \vec{v} \cdot \nabla_x f(t, \vec{x}, \vec{v}) = I[f](t, \vec{x}, \vec{v}) \tag{2.2}$$

Where $I[f]$ is referred to as the collision operator. The explicit form of $I[f]$ depends on the properties of the gas. To describe it explicitly, we make several assumptions. First, the molecules of the gas composed entirely of a single species. Second, we

assume hard sphere collisions as described in section 2.1.

2.1 Binary Collisions of Particles

This section considers the properties of two particles on a collision path with each other as illustrated in Figure 2.1. In all the work that follows, is is assumed that the molecules undergo elastic hard sphere collisions. In Figure 2.1, we model a particle

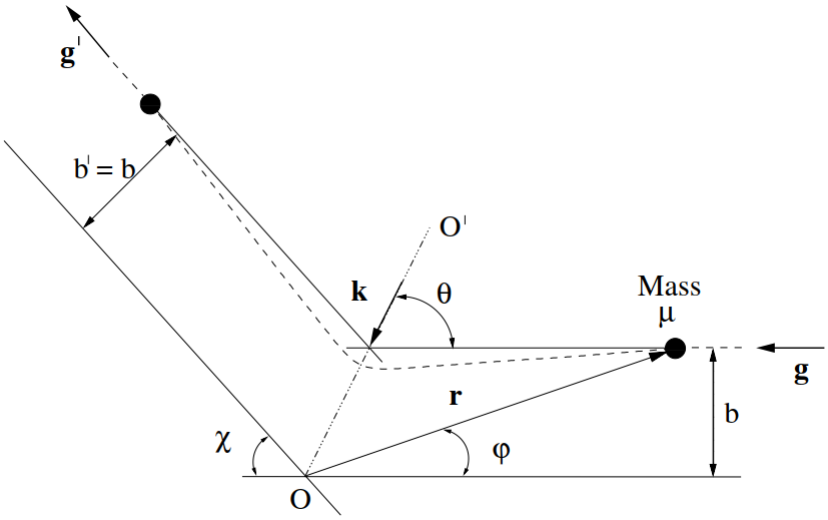

Figure 2.1: Kremer (2010, p. 27), Fig 1.6

at the point O. The reference frame is taken with respect to this particle. A second particle is approaching the particle at poisition O symbolized by the black dot. It follows the dotted line and collides with the particle and is deflected at an angle of θ and leaves the frame. Denote the pre- and post-collisional asymptotic velocities by $\vec{v}$, $\vec{v_1}$ and $\vec{v}'$, $\vec{v_1}'$ respectively. Define the relative pre- and post-collisional velocities, respectively, by

$$\vec{g} = v_1 - v\,, \quad \vec{g}' = \vec{v}' - \vec{v_1}'\,.$$

b denotes the offset of the centers of the molcules orthogonal to $\vec{g}$. ε denotes the azimuthal angle between the two particles.

By conservation of momentum, we have that

$$m\vec{v} + m\vec{v}_1 = m\vec{v}' + m\vec{v}_1' \,. \tag{2.3}$$

Equation 2.3 yields $|g'| = |g|$. In addition, due to the hard sphere assumption, the collision is considered to be perfectly elastic, giving

$$\frac{1}{2}m|\vec{v}| + \frac{1}{2}m|\vec{v}_1| = \frac{1}{2}m|\vec{v}'| + \frac{1}{2}m|\vec{v}_1'| \tag{2.4}$$

The apsidal vector, $\vec{k}$ given by

$$\vec{k} = \frac{\vec{g} - \vec{g}'}{|\vec{g} - \vec{g}'|} \,,$$

bisects the angle between asymptotic relative velocities. Using this vector, we can write a relationship between the pre- and post-collisional velocities by

$$\vec{v}_1' = \vec{v}_1 - \vec{k}(\vec{k} \cdot \vec{g}) \,, \quad \vec{v}' = \vec{v} + \vec{k}(\vec{k} \cdot \vec{g}) \,. \tag{2.5}$$

2.2 The Collision Operator

In order to explicitly write the collision operator, we must describe the collisions within the gas. Consider a particle with velocity $\vec{v}_1$ at point $\vec{x}$. This particle will collide with $f(t, \vec{x}, \vec{v}) d\vec{v}_1 g \Delta t b \, db \, d\varepsilon$ particles within the ranges of $\vec{v}_1$ and $\vec{v}_1 + d\vec{v}_1$. Integrating this by all all possible angles ($0 \leq \varepsilon < 2\pi$), over all impact parameters, ($0 \leq b \leq b^*$), and over all velocities, for all particles within the volume element $\vec{x} d\vec{x}$, we get that there are

$$dt \int_{\mathbb{R}^3} \int_0^{b^*} \int_0^{2\pi} f(t, \vec{x}, \vec{v}_1) f(t, \vec{x}, \vec{v}) b|g| \, db \, d\varepsilon \, d\vec{v}_1 \tag{2.6}$$

total collisions within the volume element that annihilate points with velocity $\vec{v}$. Likewise, there are collisions that create points with velocity $\vec{v}$. Using the results of the lasts section, points with velocity $\vec{v}'$ and $\vec{v}_1'$, impact parameter $b' = b$, and azimuthal angle $\varepsilon' = \pi + \varepsilon$ will result in particles with post-collisional velocity of $\vec{v}$ and $\vec{v}_1$. The number of such collisions is

$$\begin{aligned} & dt \int_{\mathbb{R}^3} \int_0^{b^*} \int_0^{2\pi} f(t,\vec{x},\vec{v}_1') f(t,\vec{x},\vec{v}') b' |g'| \, db \, d\varepsilon' \, d\vec{v}_1 \\ =& dt \int_{\mathbb{R}^3} \int_0^{b^*} \int_0^{2\pi} f(t,\vec{x},\vec{v}_1') f(t,\vec{x},\vec{v}') b |g| \, db \, d\varepsilon \, d\vec{v}_1 \, . \end{aligned} \tag{2.7}$$

Subtracting (2.6) from (2.7) gives the net number of particles enter or leave the volume element $\vec{v} + d\vec{v}$; that is

$$I[f](t,\vec{x},\vec{v}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} (f_1' f' - f_1 f) |g| \, b \, db \, d\varepsilon \, d\vec{v}_1 \, , \tag{2.8}$$

where,

$$f_1' \equiv f(t,\vec{x},\vec{v}_1') \quad f' \equiv f(t,\vec{x},\vec{v}') \quad f_1 \equiv f(t,\vec{x},\vec{v}_1 \quad f \equiv f(t,\vec{x},\vec{v}) \, .$$

From here, we can substitute (2.8) into (2.2) to arrive to the explicit form of the Boltzmann equation,

$$\frac{\partial}{\partial t} f(t,\vec{x},\vec{v}) + \vec{v} \cdot \nabla_x f(t,\vec{x},\vec{v}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} (f_1' f' - f_1 f) |g| \, b \, db \, d\varepsilon \, d\vec{v}_1 \, . \tag{2.9}$$

The Boltzmann equation is a non-linear integro-differential equation. The right hand side is a five dimensional integral that must be evaluated at each point in 6-dimensional space.

2.3 Moments of the Distribution Function

A gas is usually described by its macroscopic states. Kinetic theory defines these macroscopic properties in terms of distribution function $f(t, \vec{x}, \vec{v})$. The first five moments of the gas are defined below.

$$n(t, \vec{x}) = \int_{\mathbb{R}^3} f(t, \vec{x}, \vec{v}) d\vec{v} \qquad \text{- number density} \tag{2.10}$$

$$\bar{v}_i(t, \vec{x}) = \frac{1}{n(t, \vec{x})} \int_{\mathbb{R}^3} m v_i f(t, \vec{x}, \vec{v}) d\vec{v} \qquad \text{- bulk velocity} \tag{2.11}$$

$$T(t, \vec{x}) = \frac{1}{3Rn(t, \vec{x})} \int_{\mathbb{R}^3} m C^2 f(t, \vec{x}, \vec{v}) d\vec{v} \qquad \text{- temperature} \tag{2.12}$$

where $C_i = v_i - \bar{v}_i$, $C^2 = C_1^2 + C_2^2 + C_3^2$, and R is the specific gas constant. The number density $n(t, \vec{x})$ denote the number of particles contained in our distribution. $\vec{\bar{v}}(t, \vec{x})$ denotes that average velocity of particles within the gas. The temperature denotes the deviation form the average.

An important property of the collision integral is that the first five moments are conservative (see [11]), that is

$$\begin{aligned} \int_{\mathbb{R}^3} I[f](t, \vec{x}, \vec{v}) d\vec{v} &= 0\,, \\ \int_{\mathbb{R}^3} v_i I[f](t, \vec{x}, \vec{v}) d\vec{v} &= 0\,, \\ \int_{\mathbb{R}^3} C^2 I[f](t, \vec{x}, \vec{v}) d\vec{v} &= 0\,. \end{aligned} \tag{2.13}$$

2.4 The Maxwellian Distribution

If the gas is free from external influence, then the gas will approach an equilibrium. In this equilibrium, the gas distribution takes a specific shape known as the

Maxwellian distribution given below.

$$f_M(\vec{v}, n, \vec{\vec{v}}, T) = \frac{1}{\sqrt{2\pi RT}^3} \exp\left(-\frac{|\vec{v} - \vec{\vec{v}}|^2}{2RT}\right) \tag{2.14}$$

Note that the exact shape of the Maxwellian distribution depends on the macroscopic moments of the gas distribution, n, $\vec{\vec{v}}$, and T given by equations (2.10), (2.11), (2.12) respectively. This is illustrated in Figure 2.2. The distribution is centered around $\vec{\vec{v}}$. The temperature, T, controls the width of the distribution. n determines the area under the curve. The Maxwellian takes a bell shaped, where $\lim_{|\vec{v}|\to\infty} f_M(\vec{v}, n, \vec{\vec{v}}, T) = 0$, and in fact drops off rapidly as we move away from $\vec{\vec{v}}$. This will become important later when we discretize f_M (see sec **reference needed**).

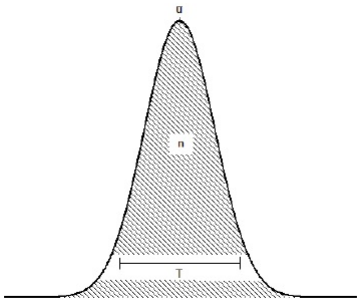

Figure 2.2: A 1D Maxwellian distribution.

When the gas is in equilibrium, the difference between the number of particles that enter and leave a particular phase volume vanishes. In other words,

$$I[f_M](t, \vec{x}, \vec{v}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} (f'_{M1} f'_M - f_{M1} f_M) g\, b\, db\, d\varepsilon\, d\vec{v}_1 \ = 0$$

2.5 Dimensionless Reduction

Gas dynamic constants can vary greatly in scale. This can cause an accumulation of round-off error using floating point arithmetic when performing a large number of computations. In order to reduce the error that can be introduced by the varying scales, a dimensionless reduction is performed on the constants and equations. The dimensionless reduction aims to reduce the scale of all the variables to roughly of order one to minimize the round-off error. Note that the dimensionless reduction process can never eliminate round-off error since it is inherit to floating point arithmetic. The techniques used for dimensionless reduction vary from problem to problem. This thesis adopts the convention borrowed from Chapter 3 of [][][][].

Let $\hat{t}$, $\hat{x}$, and $\hat{v}$ denote the conventional dimensional variables. In general, all quantities bearing $\hat{\cdot}$ will represent dimensional quantities, i.e. whose numbers have units understood to have physical meaning (e.g. seconds, meters, meters per second). For example, $\hat{f}(\hat{t}, \hat{x}, \hat{v})$ will represent the molecular number density distribution function.

We assume some time scale $\mathbb{T}$, reference temperature T_∞, and some length scale L, which are selected with a particular application in mind. We define $C_\infty = \sqrt{2RT_\infty}$. In addition, we define

$$t = \frac{\hat{t}}{\mathbb{T}}, \quad x_i = \frac{\hat{x}}{L}, \quad v = \frac{\hat{v}}{C_\infty}, \quad \text{or} \quad \hat{t} = t\mathbb{T}, \quad \hat{x} = xL, \quad \hat{v} = vC_\infty\,. \tag{2.15}$$

The dimensionless density is

$$f(t, x, v) = \frac{L^3 C_\infty^3}{N} \hat{f}(t\mathbb{T}, xL, vC_\infty) = \hat{f}(\hat{t}, \hat{x}, \hat{v})\,, \tag{2.16}$$

where N is total number of molecules in the gas volume L^3.

With these definitions, then the relationship between the dimensionless macropa-

rameters and the dimensional pacroparameters are now as follows. We define

$$\begin{aligned} n(t,x) &:= \int_{\mathbb{R}^3} f(t,x,v)dv\,, \\ n(t,x)\bar{u}(t,x) &:= \int_{\mathbb{R}^3} vf(t,x,v)dv\,, \\ n(t,x)T(t,x) &:= \frac{2}{3}\int_{\mathbb{R}^3} (v-\bar{u})^2 f(t,x,v)dv\,, \end{aligned} \tag{2.17}$$

From the above definitions, we get the relationships of the dimensionless and dimensional macroparameters. First note that in the discussion of follows, we have $d\hat{v} = C_\infty^3 dv$. Then,

$$\begin{aligned} n(t,x) &= \int_{\mathbb{R}^3} f(t,x,v)dv \\ &= \int_{\mathbb{R}^3} \frac{L^3 C_\infty^3}{N} \hat{f}(\hat{t},\hat{x},\hat{v})\frac{d\hat{v}}{C_\infty^3} \\ &= \frac{L^3}{N}\int_{\mathbb{R}^3} \hat{f}(\hat{t},\hat{x},\hat{v})d\hat{v} \\ &= \frac{L^3}{N}\hat{n}(t\mathbb{T}, xL)\,. \end{aligned} \tag{2.18}$$

In addition,

$$\begin{aligned} n(t,x)\bar{u}_i(t,x) &= \int_{\mathbb{R}^3} v_j f(t,x,v)dv \\ &= \int_{\mathbb{R}^3} \frac{L^3 C_\infty^3}{N}\frac{\hat{v}_j}{C_\infty}\hat{f}(\hat{t},\hat{x},\hat{v})\frac{d\hat{v}}{C_\infty^3} \\ &= \frac{L^3}{C_\infty N}\int_{\mathbb{R}^3} \hat{v}_j\hat{f}(\hat{t},\hat{x},\hat{v})d\hat{v} \\ &= \frac{L^3}{C_\infty N}\hat{n}(t,x,v)\hat{\bar{u}}(\hat{t},\hat{x})d\hat{v} \\ &= \frac{\hat{\bar{u}}_j(t,x)}{C_\infty}n(t,x)\,, \end{aligned} \tag{2.19}$$

and

$$
\begin{aligned}
n(t,x)T(t,x) &= \frac{2}{3}\int_{\mathbb{R}^3}(v-\bar{u})^2 f(t,x,v)dv \\
&= \frac{2}{3}\int_{\mathbb{R}^3}\frac{L^3C_\infty^3}{N}\left(\left(\frac{\hat{v}}{C_\infty}\right)^2-\left(\frac{\hat{\bar{u}}}{C_\infty}\right)^2\right)\hat{f}(\hat{t},\hat{x},\hat{v}\frac{d\hat{v}}{C_\infty^3} \\
&= \frac{2L^3}{3C_\infty^2 N}\int_{\mathbb{R}^3}(\hat{v}-\hat{\bar{u}})^2\hat{f}(\hat{t},\hat{x},\hat{v})d\hat{v} \\
&= \frac{L^3}{T_\infty N}\hat{n}(\hat{t},\hat{x})T(\hat{t},\hat{x}) \\
&= n(t,x)\frac{\hat{T}(\hat{t},\hat{x})}{T_\infty}\,.
\end{aligned}
\tag{2.20}
$$

To summarize the above results,

$$
\begin{aligned}
n(t,x) &= \frac{L^3}{N}\hat{n}(\hat{t},\hat{x})\,, \\
\bar{u}(t,x) &= \frac{\hat{\bar{u}}(\hat{t},\hat{x})}{C_\infty}\,, \\
T(t,x) &= \frac{\hat{T}(\hat{t},\hat{x})}{T_\infty}\,.
\end{aligned}
$$

Next, the Maxwellian distribution with density $\hat{n}(\hat{t},\hat{x})$, average velocity $\hat{\bar{u}}_j(\hat{t},\hat{x})$ and temperature $\hat{T}(\hat{t},\hat{x})$ translates into the dimensionless Maxwellian distribution as follows:

$$
\begin{aligned}
\hat{f}_M(\hat{t},\hat{x},\hat{u}) &= \hat{n}(2\pi R\hat{T})^{-3/2}\exp\left(-\frac{(\hat{u}-\hat{\bar{u}})^2}{2R\hat{T}}\right) \\
&= \hat{n}(2\pi RT)^{-3/2}(T_\infty/\hat{T})^{3/2}\exp\left(-\frac{(\hat{u}-\hat{\bar{u}})^2}{2RT_\infty}\frac{T_\infty}{\hat{T}}\right) \\
&= \hat{n}\pi^{-3/2}C_\infty^{-3}(1/T)^{3/2}\exp\left(-\frac{(\hat{u}-\hat{\bar{u}})^2}{T}\right) \\
&= n\frac{N}{L^3C_\infty^3}(\pi T)^{-3/2}\exp\left(-\frac{(\hat{u}-\hat{\bar{u}})^2}{T}\right) \\
&= \frac{N}{L^3C_\infty^3}f_M(t,x,u)\,,
\end{aligned}
\tag{2.21}
$$

where

$$f_M(t,x,u) := \frac{n}{(\pi T)^{3/2}} \exp\Big(-\frac{(u-\bar{u})^2}{T}\Big)\,. \tag{2.22}$$

Chapter 3

Discontinuous Galerkin Discretization in the Velocity Variable

Discontinuous Galerkin methods is a method of discretizing equations. The following section describes the DG formulation found in [1], [3].

3.1 DG Discretization in Velocity Space

We denote the points in the velocity space as $\vec{v} = (u, v, w)$. The velocity space is reduced to a rectangular parallelpiped $K = [u_L, u_R] \times [v_L, v_R] \times [w_L, w_R]$. It is assumed that outside the parallelpiped the contribution of the function to the first few moments is negligible. Depending on the parallelpiped, this will not result in large errors in terms of conservation of density and temperature.

We partition K into N smaller rectangular parallelpipeds $K_j = [u_L^j, u_R^j] \times [v_L^j, v_R^j] \times [w_L^j, w_R^j]$. Each K_j will contain a set of basis function, ϕ_j^i, $i = 1, \ldots, s$ as described. We introduce nodes of Gauss quadratures of order s_u, s_v, and s_w on each of the intervals $[u_L^j, u_R^j]$, $[v_L^j, v_R^j]$, and $[w_L^j, w_R^j]$. The nodes are denoted as $\kappa_{p;j}^u$, $\ldots$, $p = 1, s_u$, $\kappa_{q;j}^v$, $\ldots$, $q = 1, s_v$, and $\kappa_{r;j}^w$, $\ldots$, $r = 1, s_w$. From the nodes, the one-dimensional Lagrange basis functions are defined:

$$\phi_{l;j}^u(u) = \prod_{\substack{p=1,s_u \\ p \neq l}} \frac{\kappa_{p;j}^u - u}{\kappa_{p;j}^u - \kappa_{l;j}^u}, \quad \phi_{m;j}^v(v) = \prod_{\substack{q=1,s_v \\ q \neq m}} \frac{\kappa_{q;j}^v - v}{\kappa_{q;j}^v - \kappa_{m;j}^v}, \quad \phi_{n;j}^w(w) = \prod_{\substack{r=1,s_w \\ r \neq n}} \frac{\kappa_{r;j}^w - w}{\kappa_{r;j}^w - \kappa_{n;j}^w}. \tag{3.1}$$

The three-dimensional basis function is defined as

$$\phi_{i;j}(\vec{v}) = \phi_{l;j}^u(u)\phi_{m;j}^v(v)\phi_{n;j}^w(w) \tag{3.2}$$

where $l = 1, \ldots, s_u$, $m = 1, \ldots, s_v$, $n = 1, \ldots, s_w$ and i is the index that runs through

all possible combinations of l, n, and m, and is computed as $i = ((l-1)s_v) + (m-1)s_w) + n$. A useful property of the basis functions (3.2) is that they vanish on all nodes except one. In addition, the quadrature nodes used are exact on polynomials of degree at most $2s_u - 1$, $2s_v - 1$, and $2s_w - 1$. In addition, the following lemma holds,

Lemma 1 *(see also [2, 8]) The following identities hold for basis functions $\phi_{i;j}(\vec{v})$:*

$$\int_{K_j} \phi_{p;j}(\vec{v})\phi_{q;j}(\vec{v})\, d\vec{v} = \frac{\omega_p \Delta\vec{v}^j}{8}\delta_{pq} \qquad \textit{and} \qquad \int_{K_j} \vec{v}\phi_{p;j}(\vec{v})\phi_{q;j}(\vec{v})\, d\vec{v} = \frac{\omega_p \Delta\vec{v}^j}{8}\vec{v}_{p;j}\delta_{pq}, \tag{3.3}$$

where indices l, n, and m of one dimensional basis functions correspond to the three-dimensional basis functions $\phi_{p;j}(\vec{v}) = \phi^u_{l;j}(u)\phi^v_{m;j}(v)\phi^w_{n;j}(w)$, and the vector $\vec{v}_{p;j} = (\kappa^u_{l;j}, \kappa^v_{m;j}, \kappa^w_{n;j})$.

3.2 Nodal-DG Velocity Discretization of the Boltzmann Equation

We assume that on each K_j, the solution to the Boltzmann equation is sought of the form

$$f(t, \vec{x}, \vec{v})|_{K_j} = \sum_{i=1,s} f_{i;j}(t, \vec{x})\phi_{i;j}(\vec{v})\,. \tag{3.4}$$

We substitute equation 3.4 into 2.2, multiply the result by test basis function, integrate over K_j, and apply identity (3.3) to arrive to

$$\partial_t f_{i;j}(t, \vec{x}) + \vec{v}_{i;j} \cdot \nabla_x f_{i;j}(t, \vec{x}) = \frac{8}{\omega_i \Delta\vec{v}^j} I_{\phi_{i;j}}\,, \tag{3.5}$$

where $I_{\phi_{i;j}}$ is the projection of the collision operator on the basis function $\phi_{i;j}(\vec{v})$:

$$I_{\phi_{i;j}} = \int_{K_j} \phi_{i;j}(\vec{v}) I[f](t, \vec{x}, \vec{v})\, d\vec{v}\,. \tag{3.6}$$

3.3 Reformulation of the Galerkin Projection of the Collision Operator

Similarly to [1, 2, 12], we rewrite the DG projection of the collision operator $I_{\phi_{i;j}}$ in the form of a bilinear integral operator with a time-independent kernel. The principles of kinetic theory suggest that changes to $f(t, \vec{x}, \vec{v})$ with respect to $\vec{x}$ at the distance of a few b^* are negligible, see e.g., [15]. Specifically, using the well-known identities (see, e.g., [10], Section 2.4), and applying the first principles assumption, we have

$$\begin{aligned} I_{\phi_{i;j}} &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v}) f(t,\vec{x},\vec{v}_1) \int_{\mathbb{S}^2} (\phi_{i;j}(\vec{v}') - \phi_{i;j}(\vec{v})) b_\alpha(\theta) |g|^\alpha \, d\sigma \, d\vec{v}_1 \, d\vec{v} \\ &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v}) f(t,\vec{x},\vec{v}_1) A(\vec{v},\vec{v}_1;\phi_{i;j}) d\vec{v}_1 \, d\vec{v} \,, \end{aligned} \tag{3.7}$$

where

$$A(\vec{v},\vec{v}_1;\phi_{i;j}) = |g|^\alpha \int_{\mathbb{S}^2} (\phi_{i;j}(\vec{v}') - \phi_{i;j}(\vec{v})) b_\alpha(\theta) \, d\sigma \,. \tag{3.8}$$

The kernel $A(\vec{v},\vec{v}_1;\phi_{i;j})$ is independent of time and can be pre-computed. In [2] properties of a kernel closely related to $A(\vec{v},\vec{v}_1;\phi_{i;j})$ are considered. In particular, due to the local support of $\phi_{i;j}(\vec{v})$, it is anticipated that kernel $A(\vec{v},\vec{v}_1;\phi_{i;j})$ will have only $O(M^5)$ non-zero components for each $\phi_{i;j}(\vec{v})$, where M is the number of discrete velocity points in each velocity dimension. As a result, evaluation of (3.7) will require $O(M^8)$ operations for each spatial point. This number of evaluations is very high. However, as we will show later, it can be reduced to $O(M^6)$ operations using symmetries of $A(\vec{v},\vec{v}_1;\phi_{i;j})$, the convolution form of (3.7) and the Fourier transform.

We remark that in many numerical re-formulations of the Boltzmann equation, the collision operator is separated into the gain and loss terms. This separation can

be performed in (3.7),

$$
\begin{aligned}
I_{\phi_{i;j}} &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1)\int_{\mathbb{S}^2}(\phi_{i;j}(\vec{v}')-\phi_{i;j}(\vec{v}))b_\alpha(\theta)|g|^\alpha\, d\sigma\, d\vec{v}_1\, d\vec{v} \\
&= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1)(\int_{\mathbb{S}^2}\phi_{i;j}(\vec{v}')b_\alpha(\theta)|g|^\alpha d\sigma\ - |g|^\alpha\int_{\mathbb{S}^2}\phi_{i;j}(\vec{v})b_\alpha(\theta)d\sigma\ )d\vec{v}_1\, d\vec{v} \\
&= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1)(A^+(\vec{v},\vec{v}_1;\phi_{i;j})-|g|^\alpha\sigma_T)d\vec{v}_1\, d\vec{v} \qquad (3.9)
\end{aligned}
$$

where

$$
A^+(\vec{v},\vec{v}_1;\phi_{i;j}) = \int_{\mathbb{S}^2}\phi_{i;j}(\vec{v}')b_\alpha(\theta)|g|^\alpha d\sigma\,, \quad \sigma_T = \int_{\mathbb{S}^2}\phi_{i;j}(\vec{v})b_\alpha(\theta)d\sigma\,. \qquad (3.10)
$$

$A^+(\vec{v},\vec{v}_1;\phi_{i;j})$ has similar properties to that of $A(\vec{v},\vec{v}_1;\phi_{i;j})$, but $A^+(\vec{v},\vec{v}_1;\phi_{i;j})$ has certain properties that are better than that of $A(\vec{v},\vec{v}_1;\phi_{i;j})$ for a Fourier transform. $A(\vec{v},\vec{v}_1;\phi_{i;j})$ is grows linearly to infinity in some direction whereas $A^+(\vec{v},\vec{v}_1;\phi_{i;j})$ does not. It can then be argued that $A^+(\vec{v},\vec{v}_1;\phi_{i;j})$ is better suited for a Fourier transform than $A(\vec{v},\vec{v}_1;\phi_{i;j})$. The algorithms in this paper also have straightforward extensions to the the split formulation of (3.7). However, in practice, the split formulation of (3.7) had significant errors in conservation. The exact mechanism of why the non-split formulation preserves the conservation laws better is still not clear to the author. Some insight can be obtained by noticing that values of the collision kernel $A(\vec{v},\vec{v}_1;\phi_{i;j})$ span several orders of magnitude. Small values of $A(\vec{v},\vec{v}_1;\phi_{i;j})$ occur in sufficiently many points so that they are important collectively. It is possible that when small and large values are combined during the evaluation of the gain term, the accuracy of the small values is lost or essentially diminished. When the loss term is subtracted from the gain term, cancellation occurs producing large errors. On the contrary, conservation laws are satisfied point-wise in the form (3.7), (3.8) up to a small number of algebraic manipulations with the basis functions $\phi_{i;j}(\vec{v})$. Because of these considerations, we chose to use the non-split form of the collision operator in

simulations.

3.4 Properties of the Kernel $A(\vec{v}, \vec{v}_1; \phi_{i;j})$

As stated in section 3.3, $A(\vec{v}, \vec{v}_1; \phi_i^j)$ only contains $\mathcal{O}(M^5)$ non-zero components. This section will justify that claim by going over properties of $A(\vec{v}, \vec{v}_1; \phi_i^j)$.

Lemma 2 *Let $A(\vec{v}, \vec{v}_1; \phi_i^j)$ be defined by (3.8) with all gas particles having the same mass with the particle potentials being spherically symmetric. Then $A(\vec{v}, \vec{v}_1; \phi_i^j)$ is spherically symmetric with respect to $\vec{v}$ and $\vec{v}_1$, that is*

$$A(\vec{v}, \vec{v}_1; \phi_i^j) = A(\vec{v}_1, \vec{v}; \phi_i^j),\ \forall \vec{v}, \vec{v}_1 \in \mathbb{R}^3. \tag{3.11}$$

Also,

$$A(\vec{v}, \vec{v}) = 0, \forall \vec{v} \in \mathbb{R}^3 \tag{3.12}$$

The next lemma establishes the shift-invariance property of $A(\vec{v}, \vec{v}_1; \phi_i^j)$ within the velocity space.

Lemma 3 *Let operator $A(\vec{v}, \vec{v}_1; \phi_{i;j})$ be defined by (3.8). Then $\forall \xi \in \mathbb{R}^3$*

$$A(\vec{v} + \vec{\xi}, \vec{v}_1 + \vec{\xi}; \phi_{i;j}(\vec{v} - \vec{\xi})) = A(\vec{v}, \vec{v}_1; \phi_{i;j})\,.$$

Proof. Consider $A(\vec{v}+\vec{\xi}, \vec{v}_1+\vec{\xi}; \phi_{i;j}(\vec{v}-\vec{\xi}))$. We clarify that these notations mean that particle velocities $\vec{v}$ and $\vec{v}_1$ in (3.8) are replaced with $\vec{v}+\vec{\xi}$ and $\vec{v}_1+\vec{\xi}$ correspondingly and that basis function $\phi_{i;j}(\vec{v})$ is replaced with a "shifted" function $\phi_{i;j}(\vec{v}-\vec{\xi})$. We notice that the relative speed of the molecules with velocities $\vec{v}+\vec{\xi}$ and $\vec{v}_1+\vec{\xi}$ is still $\vec{g} = \vec{v}+\vec{\xi}-(\vec{v}_1+\vec{\xi}_1) = \vec{v}-\vec{v}_1$. The post-collision velocities for the pair of particles will be $\vec{v}'+\vec{\xi}$ and $\vec{v}_1'+\vec{\xi}$, where $\vec{v}'$ and $\vec{v}_1'$ are given by (2.5). We notice, in particular, that

choices of θ and ε in (2.5) are not affected by $\vec{\xi}$. The rest of the statement follows by a direct substitution:

$$\begin{aligned} A(\vec{v}+\vec{\xi},\vec{v}_1+\vec{\xi};\phi_{i;j}(\vec{v}-\vec{\xi})) \\ &= |g|^\alpha \int_{\mathbb{S}^2} \phi_{i;j}((\vec{v}'+\vec{\xi})-\vec{\xi}) b_\alpha(\theta)\, d\sigma \\ &= |g|^\alpha \int_{\mathbb{S}^2} \phi_{i;j}(\vec{v}') b_\alpha(\theta)\, d\sigma \\ &= A(\vec{v},\vec{v}_1;\phi_{i;j})\,. \end{aligned}$$

■

We remark that Lemma 3 holds for all potentials of molecular interaction used in rarefied gas dynamics. This property was used in [2] to reduce the storage requirement on uniform partitions. In the case of uniform partitions with the same basis element on every cell, Information about $A(\vec{v},\vec{v}_1;\phi_{i;j})$ need only be stored for a single cell. Other values of $A(\vec{v},\vec{v}_1;\phi_{i;j})$ can be restored using the shift invariance. Strictly speaking, one requires an infinite partition of the entire velocity space to recover all points. However, assuming that the support of the solution is well contained within a finite domain, the shift invariance property can be used successfully.

3.5 Rewriting the Collision Operator in the Form of a Convolution

It was shown in [3] that the Galerkin projection of the collision operator can be reformulated in terms of a convolution. This work is recalled in this section. We select a partition cell K_c and designate this cell as a generating cell. Similarly, the basis functions $\phi_{i;c}(\vec{v})$ on K_c are designated as the generating basis functions. Basis functions $\phi_{i;j}(\vec{v})$ on other cells can be obtained using a shift in the velocity variable, namely $\phi_{i;j}(\vec{v}) = \phi_{i;c}(\vec{v}+\vec{\xi}_j)$ where $\vec{\xi}_j \in \mathbb{R}^3$ is the vector that connects the center of K_j to the center of K_c.

According to Lemma 3, operator $A(\vec{v}, \vec{v}_1, \phi_{i;j})$ is invariant with respect to translations. Therefore

$$\begin{aligned} I_{\phi_{i;j}} &= \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} f(t, \vec{x}, \vec{v}) f(t, \vec{x}, \vec{v}_1) A(\vec{v} + \vec{\xi_j}, \vec{v}_1 + \vec{\xi_j}; \phi_{i;j}(\vec{u} - \vec{\xi_j}))\, d\vec{v}_1 d\vec{v} \\ &= \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} f(t, \vec{x}, \vec{v}) f(t, \vec{x}, \vec{v}_1) A(\vec{v} + \vec{\xi^j}, \vec{v}_1 + \vec{\xi_j}; \phi_{i;c}(\vec{u}))\, d\vec{v} d\vec{v_1}\,. \end{aligned} \tag{3.13}$$

Performing the substitutions $\vec{\hat{v}} = \vec{v} + \vec{\xi_j}$ and $\vec{\hat{v}}_1 = \vec{v}_1 + \vec{\xi_j}$ in (3.13), we have

$$I_{\phi_{i;j}} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} f(t, \vec{x}, \vec{\hat{v}} - \vec{\xi_j}) f(t, \vec{x}, \vec{\hat{v}}_1 - \vec{\xi_j}) A(\vec{\hat{v}}, \vec{\hat{v}}_1; \phi_{i;c}(\vec{u}))\, d\vec{\hat{v}} d\vec{\hat{v}}_1\,.$$

We then introduce a bilinear convolution operator, $i = 1, \ldots, s$

$$I_i(\vec{\xi}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} f(t, \vec{x}, \vec{v} - \vec{\xi}) f(t, \vec{x}, \vec{v}_1 - \vec{\xi}) A(\vec{v}, \vec{v}_1; \phi_{i;c})\, d\vec{v} d\vec{v}_1\,, \tag{3.14}$$

and notice that $I_{\phi_{i;j}}$ can be obtained from (3.14) as $I_{\phi_{i;j}} = I_i(\vec{\xi_j})$. In the following, we will refer to (3.14) as the convolution form of the Galerkin projection of the collision integral.

3.6 Discretization of the Collision Integral

In order to calculate (3.14), we replace the three-dimensional integrals with the Gauss quadratures associated with then nodal-DG discretization, (3.2). As is discussed above, we are only interested in computing convolution (3.14) at vectors $\vec{\xi} = \vec{\xi_j}$ that connect centers of the velocity cells K_j to the center of the velocity cell K_c, the support of $\phi_{i;c}(\vec{v})$. Since the same nodal points are used on all velocity cells, shifts $\vec{\xi_j}$ translate nodal points in one cell to nodal points in another cell. As a result, the quadrature sums to evaluate convolution (3.14) use values of the unknown $f(t, \vec{x}, \vec{v})$ at the nodal points only. In fact, the shift in the velocity variable $\vec{v}_{i;l} - \vec{\xi_j}$ will correspond to a shift in the three dimensional index of the velocity cell which we will

write formally as $l-j$, producing the velocity node $\vec{v}_{i;l-j}(\vec{v})$. The exact expression for the shift $l-j$ will be made clear later by considering the cell indices in each velocity dimension. The index i of the node within the cell is not affected in this process.

We can write the discrete form of (3.14) as

$$I_{i;j} := I_i(\vec{\xi}_j) = \sum_{i',i''=1}^{s} \sum_{j'=1}^{M^3} \sum_{j''=1}^{M^3} f_{i';j'-j} f_{i'';j''-j} A_{i',i'';j',j'';i} \tag{3.15}$$

where $f_{i';j'-j} = f(t, \vec{x}, \vec{v}_{i';j'-j})$, $A_{i',i'';j',j'';i} = A(\vec{v}_{i';j'}, \vec{v}_{i'';j''}; \phi_{i;c})$ and the three dimensional indices i' and i'' run over the velocity nodes within a single velocity cell and indices j' and j'' run over all velocity cells. We note that for some index shifts $j'-j$, the resulting cells are outside of the velocity domain. In [2] the values outside of the domain were substituted with zeros. In cases when the support of the solution was well contained within the computational domain, this assumption did not lead to large numerical errors.

We note that in order to calculate (3.15), it would require $O(M^9)$ operations to calculate it, $O(M^6)$ at one velocity node for $O(M^3)$ velocity nodes. However, recall the property stated in section 3.3 that A only has $O(M^5)$ components bringing down the total cost of computing (3.15) down to $O(M^8)$ operations. However, we wish to reduce this further. It turns out that the Discrete Fourier Transform has properties that allow us to bring down the cost of computing (3.15) from $O(M^8)$ to $O(M^6)$. The DFT and its properties are discussed in the next section.

3.7 The Micro-Macro Decomposition

The form of the collision integral can be reformulated for the purpose of numerical implementation. During simulations, it was observed that the ability of numerical simulations to conserve mass, momentum, and energy is strongly affected by the form of the discrete collision integral.

One such reformulation is the micro-macro decomposition [**??**] which considers the solution has the sum of the target Maxwellian distribution and the deviation from the Maxwellian, i.e.,

$$f(t,\vec{x},\vec{v}) = f_M(t,\vec{x},\vec{v}) + \Delta f(t,\vec{x},\vec{v})\,, \tag{3.16}$$

where $f_M(t,\vec{x},\vec{v})$ takes the form (2.14) with the same temperature, bulk velocity, and temperature as $f(t,\vec{x},\vec{v})$. This form was applied in [1] in order to improve conservation properties of the scheme when the solution is near continuum. The numerical implementation used for the FFT convolution also used this decomposition. We substitute (3.16) into (3.7) to obtain an alternative form of the collision integral:

$$\begin{aligned} I_{\phi_{i;j}} &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} f(t,\vec{x},\vec{v}) f(t,\vec{x},\vec{v}_1) A(\vec{v},\vec{v}_1;\phi_{i;j}) d\vec{v}_1\, d\vec{v} \\ &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} [f_M(t,\vec{x},\vec{v})\Delta f(t,\vec{x},\vec{v}_1) + \Delta f(t,\vec{x},\vec{v}) f_M(t,\vec{x},\vec{v}_1) \\ &\qquad + \Delta f(t,\vec{x},\vec{v})\Delta f(t,\vec{x},\vec{v}_1)] A(\vec{v},\vec{v}_1;\phi_{i;j}) d\vec{v}_1\, d\vec{v}\,. \end{aligned} \tag{3.17}$$

where we used that $I[f_M](t,\vec{x},\vec{v}) = 0$. It was found that this form of the operator provided much better numerical results. Thus the numerical results in the rest of this work use (3.17) for all the calculations.

Chapter 4

The Discrete Fourier Transform

The goal is to compute (3.15) faster than $O(M^8)$. We note that (3.15) is similar to a discrete convolution form (which will be defined later). The Discrete Fourier Transform (DFT) can be used to speed up the computations of convolutions using the *Convolution Theorem*. In particular, it is used to speed up circular convolutions. In order to apply the DFT to our problem, we must first assume a periodic extension of the discrete form of f. This section discusses the background of the Discrete Fourier Transform, convolution theorem, and periodic continuation needed to quickly compute (3.15)

4.1 The One-Dimensional Discrete Fourier Transform and its Properties

The Discrete Fourier Transform is the discrete analog of the Fourier transform. It is a tool often used describe the relationship between the time and frequency representation of discrete signals [13].

Definition 1 *Let $\{x_n\}_{n=0}^{N-1}$ be a sequence of N complex numbers. The DFT is defined as*

$$\mathcal{F}[x]_k = \sum_{l=0}^{N-1} W^{lk} x_l, \qquad \textit{where} \quad W = \mathrm{e}^{-\imath 2\pi/N}\,. \tag{4.1}$$

4.1.1 Properties of the DFT

Many properties of the DFT are a natural consequence of its defintion. In this subsection, we review the properties of the DFT that will be important for applying the DFT to 3.15. First, the DFT is invertible.

Definition 2 *Let $\{\mathcal{F}[x]_k\}_{k=0}^{N-1}$ be a sequence of N complex numbers such that it is the*

DFT of the complex valued sequence $\{x_l\}_{l=0}^{N-1}$. *Then the inverse DFT is defined as*

$$x_l = \frac{1}{N}\sum_{k=0}^{N-1} W^{-lk}\mathcal{F}[x]_k\,. \tag{4.2}$$

We can see that definition of the inverse will give back the original sequence. If we have a sequence $\{x_l\}_{l=0}^{N-1}$, then note that if we take the inverse of $\mathcal{F}[x]_k$

$$\begin{aligned}
\mathcal{F}^{-1}[\mathcal{F}[x]]_l &= \frac{1}{N}\sum_{k=0}^{N-1} W^{-lk}(\sum_{j=0}^{N-1} W^{kj}x_j) \\
&= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1} W^{-lk}W^{kj}x_j \\
&= \frac{1}{N}\sum_{j=0}^{N-1} x_j(\sum_{k=0}^{N-1} W^{(j-l)k}) \\
&= N\frac{1}{N}x_l = x_l
\end{aligned}$$

where the last step used that if $l \neq j$ then $\sum_{k=0}^{N-1} W^{(j-l)k} = 0$ and for $l = j$ we have that $\sum_{k=0}^{N-1} W^{(j-l)k} = \sum_{k=0}^{N-1} 1 = N$. Thus the definition given above returns the original sequence. To see this:

$$\begin{aligned}
\mathcal{F}[z]_k &= \sum_{l=0}^{N-1} W^{kn}\sum_{n=0}^{N-1} x_n y_{l-n} \\
&= \sum_{n=0}^{N-1} x_n(\sum_{l=0}^{N-1} W^{kn}y_{l-n})
\end{aligned}$$

The DFT is linear. That is, if we have two complex sequences $\{x_n\}_{n=0}^{N-1}$ and $\{y_n\}_{n=0}^{N-1}$, then

$$\mathcal{F}[x+y]_k = \mathcal{F}[x]_k + \mathcal{F}[y]_k\,. \tag{4.3}$$

This property will be used in order to apply the DFT to discrete convolution form of

the collision integral.

The last property we need is the shift property. This says that if $\{y_n\}$ is a periodic sequence created by shifting $\{x_n\}$ by l, that is $\{y_n\} = \{x_{n-l}\}$, then

$$\mathcal{F}[y]_k = W^{kl}\mathcal{F}[x]_k \,. \tag{4.4}$$

To see this

$$\begin{aligned}
\mathcal{F}[y]_k &= \sum_{n=0}^{N-1} W^{kn} y_n \\
&= \sum_{n=0}^{N-1} W^{kn} x_{n-l} \\
&= \sum_{n=0}^{N-1} W^{k(n-l)} x_{n-l} \\
&= W^{kl} \sum_{n=0}^{N-1} W^{k(n-l)} x_{n-l} \\
&= W^{kl}\mathcal{F}[x]_k \,.
\end{aligned}$$

4.1.2 The Convolution Theorem

Another important result is the convolution theorem. The result is stated and proved here. First we start with a definition.

Definition 3 *Let x_n and y_n be periodic sequences with period N. An N-point circular convolution of x_n and y_n is defined as (see, e.g., [13])*

$$z_l = \sum_{n=0}^{N-1} x_n y_{l-n} \,, \quad l = 1, \ldots, N-1 \,. \tag{4.5}$$

We are now ready to state the convolution theorem.

Theorem 1 *Let $\{x_n\}$ and $\{y_n\}$ be N-periodic sequences. Let z_l denote the circular convolution given by (4.5). Then*

$$\mathcal{F}[z]_k = \mathcal{F}[x]_k \mathcal{F}[y]_k \,. \tag{4.6}$$

Proof. This can be proven with a straight forward use of the definition of the DFT and the circular convolution,

$$\begin{aligned}
\mathcal{F}[z]_k &= \sum_{l=0}^{N-1} W^{kl} \left(\sum_{n=0}^{N-1} x_n y_{l-n} \right) \\
&= \sum_{n=0}^{N-1} x_n \left(\sum_{l=0}^{N-1} W^{kl} y_{l-n} \right) \\
&= \sum_{n=0}^{N-1} x_n W^{kn} \mathcal{F}[y]_k \\
&= \mathcal{F}[y]_k \left(\sum_{n=0}^{N-1} x_n W^{kn} \right) \\
&= \mathcal{F}[y]_k \mathcal{F}[x]_k \,,
\end{aligned}$$

where we used the shift property (4.4) of the DFT, e.g. $\sum_{l=0}^{N-1} W^{kl} y_{l-n} = W^{kl} \mathcal{F}[y]_k$

■

After calculating $\mathcal{F}[z]_k$ using (4.6), the sequence $\{z_n\}_{n=0}^{N-1}$ can be retrieved via application of the inverse DFT, (4.2), e.g. $\mathcal{F}^{-1}[\mathcal{F}[z]]_n = z_n$.

The convolution theorem seemingly gives no computational gain. Straightforward computation of the DFT and inverse DFT will take $O(N^2)$ complex operations. Computation of (4.5) also takes $O(N^2)$ operations without needing to perform any complex multiplications if both $\{x_n\}$ and $\{y_n\}$ are real. However, the complexity of calculating (4.1) can be reduced by an algorithm known as the Fast Fourier Transform (FFT) which can calculate both the forward and inverse DFT in $O(N \log N)$ operations.

4.2 Circular Convolution as Linear Convolution

4.2.1 Linear Convolutions and Circular Convolutions

rewrite The DFT can be used to calculate circular convolutions quickly. However, while (3.15) has a form similar to the definition of a circular convolution, we note that it is not a circular convolution. In the discussion that follows (3.15), we mention that for an $f_{i;j}$ with $j \leq 0$ or $j > M$, we set $f_{i;j} = 0$. If (3.15) was truly a circular convolution, then we must assume that $f_{i;j} = f_{i;j+M}$ for any $j \leq 0$ or $f_{i;j} = f_{i;k-M}$ for any $j > M$.

Formally, (3.15) is a linear convolution. However, this section will show that there are deep connections between linear convolutions and circular convolutions. Indeed, there are conditions where they can be considered equivalent. This allows us to treat (3.15) as circular convolution and use the convolution theorem of the DFT and the speed up provided by the Fast Fourier Transform algorithm to calculate (3.15) quickly. The discussion here is inspired by Oppenheim, Shafter and Buck Digital signal processing (add citation).

4.2.2 Linear Convolution of Two Finite-Length sequences

For the sake of ease of notation for later discussions, we introduce a linear convolution for infinite sequences.

Definition 4 *Let* $\{x_n\}_{n=-\infty}^{\infty}$, $\{y_n\}_{n=-\infty}^{\infty} \subset \mathbb{C}$ *be sequences of (possibly complex) numbers. Then a linear convolution is defined as*

$$z_l = \sum_{n=-\infty}^{\infty} x_n y_{l-n}\,, l \in \mathbb{N} \tag{4.7}$$

If the sequences are finite sequences of length L, then we may set $x_n = 0$ and

$y_n = 0$ if $n < 0$ or $n \geq L$. In that case, the above definition readily extends to finite sequences.

Suppose we have two sequences $\{x_n\}_{n=-\infty}^{\infty}$,$\{y_n\}_{n=-\infty}^{\infty}$ such that $x_n = 0$ for $n < 0$ and $n \geq L$ and $y_n = 0$ for $n < 0$ and $n \geq P$, and let z_l be the result of performing a linear convolution between these two sequences. Note that $x_n y_{l-n} = 0$ for $l < 0$ and $l > L + P - 2$. In other words, $z_l \neq 0$ for $0 \leq l \leq L + P - 2$. If $\{x_n\}$ and $\{y_n\}$ are finite sequences, then z_l is also a finite sequence of maximum length $L + P - 1$. In this, using finite sequences, we can rewrite the sum (4.7) as

$$z_l = \sum_{n=\max(0,l-(P-1))}^{\min(L-1,l)} x_n y_{l-n} \tag{4.8}$$

4.2.3 Linear Convolution as Circular Convolution

Due to the similarity of the definition of the linear convolution (4.7) and the circular convolution (4.5), there are conditions in which linear convolutions and circular convolutions are equivalent. The linear convolution involves multiplying a sequence by a index-reversed and shifted version of the other, then summing the products, i.e. $x_n y_{l-n}$ where y_n is reversed and shifted. The circular convolution also reverses and shifts, but in particular, the finite sequences of $\tilde{x}_n$ and $\tilde{y}$ are periodic. However, if we extend the sequences $\tilde{x}_n$ and $\tilde{y}_n$ in the appropriate manner, we can eliminate the difference between the linear convolution and circular convolution of sequences.

We start with a motivating example. Let's examine the case where $x_n = 1$ for $n = 0, \ldots, 4$ and $y_n = 1$ for $n = 0, \ldots, 4$. The linear convolution would be calculated as

$$z_l = \sum_{n=-\infty}^{\infty} x_n y_{l-n} = \begin{cases} x_0 y_0 & = 1, \quad l = 0 \\ x_0 y_1 + x_1 y_0 & = 2, \quad l = 1 \\ & \vdots \\ x_0 y_4 + \cdots + x_4 y_0 & = 5, \quad l = 4 \\ & \vdots \\ x_4 y_0 & = 1, \quad l = 8 \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

The linear convolution gives a triangular sequence, $z_l = \min(l+1, 9-l)$ for $l = 0, \ldots, 8$. However, if we were to directly do a circular convolution, we would simply get $\tilde{z}_l = 5$ for $l = 0, \ldots, 4$. However, let $\{\tilde{x}_n\}$ and $\{\tilde{y}\}$ be periodic sequences of period 9 such that $\tilde{x}_n = x_n$, $\tilde{y}_n = y_n$ for $n = 0, \ldots, 4$ and $\tilde{x}_n = \tilde{y} = 0$ otherwise. Let $\tilde{z}_n$ denote the sequence generated by the circular convolution of $\tilde{x}_n$ and $\tilde{y}_n$. Then note that (4.9) still holds for $\tilde{z}_n$.

The insight is that we can *pad* the sequences with zeroes such that the zeros remove any extra terms added by the circular convolution. Let z_n denote the result of linear convolution of $\{x_n\}$ and $\{y_n\}$. Suppose we have periodic sequences $\{\tilde{x}_n\}$, $\{\tilde{y}_n\}$ such that $\tilde{x}_n = x_n$ for $0 \leq n < L$ and $\tilde{x}_n = 0$ for $L \leq n \leq L+P-2$, and $\tilde{y}_n = y_n$ for $0 \leq n < P$ and $\tilde{y}_n = 0$ for $L \leq n \leq L+P-2$. Set $N = L+P-2$. Let $\tilde{z}$ denote the sequence generated by taking the N-point circular convolution of $\tilde{x}$ and $\tilde{y}$. Then we have that $\tilde{z}_n = z_n$.

To see this, note that for $l = 0$, we have that N-point convolution becomes $\sum_{n=0}^{L+P-2} \tilde{x}_n \tilde{y}_{l-n} = \tilde{x}_0 \tilde{y}_0$ since $\tilde{x}_1, \ldots \tilde{x}_{L-1}$ is going to be multiplied against $\tilde{y}_{L+P-2}, \ldots, \tilde{y}_{L+P-2-(L-2)}$ $= \tilde{y}_P$ which are all zero. Then for $l = 1$, we must have that $\tilde{z}_l = \sum_{n=0}^{L+P-2} \tilde{x}_n \tilde{y}_{l-n} = \tilde{x}_0 \tilde{y}_1 + \tilde{x}_1 \tilde{y}_0$ which agrees with equation (4.8). Thus we have shown how to recover

the linear convolution using the circular convolution.

Now that we have shown that we can indeed recover a linear convolution using a circular convolution with zero padding, we can now apply the convolution theorem to linear convolutions. To see this, if we set $N > L + P - 2$, so if we take an N-point circular convolution of $\{x_n\}$ and $\{y_n\}$, (or equivalently $\{\tilde{x}_n\}$ and $\{\tilde{y}_n\}$, then we have that

$$\mathcal{F}[z]_k = \mathcal{F}[x]_k \mathcal{F}[y]_k \,,$$

where $\mathcal{F}$ denotes the N-point DFT. In particular, with the correct zero padding, we can apply the DFT to (3.15) with no error.

4.2.4 Circular Convolution as Linear Convolution with Aliasing.

We saw that we can find an equivalence between the linear convolution and the circular convolution by zero padding the finite sequences $\{x_n\}$ and $\{y_n\}$. However, in certain cases, zero-padding may not be possible. In this section we explore what happens when we are unable to zero-pad the sequences.

Suppose $\{x_n\}_{n=0}^{L-1}$ and $\{y_n\}_{n=0}^{P-1}$ For simplicity, let $L \geq P$. If we take an L-point DFT of $\{x_n\}$ or $\{y_n\}$ then we can perfectly reconstruct both $\{x_n\}$ and $\{y_n\}$ given $\{\mathcal{F}[x]_k\}$ and $\{\mathcal{F}[y]_k\}$. Now suppose z_l is the linear convolution of $\{x_n\}$ and $\{y_n\}$, so z_l is of length $L + P - 1$. If we attempt to calculate $\mathcal{F}[z]_k$ using the convolution theorem, we expect it to be incorrect because an L-point DFT would be insufficient. Suppose we attempt to calculate $\mathcal{F}[z]_k$ by using the convolution theorem, i.e.

$$\mathcal{F}[z]_k^* = \mathcal{F}[x]_k \mathcal{F}[y]_k \,. \tag{4.10}$$

Then we can expect that in general $\mathcal{F}^{-1}[\mathcal{F}[z]^*]_l \neq z_l$. We refer to this as *aliasing error*.

This occurs because by trying to calculate the linear convolution using the covolution theorem, we are implicitly using a circular convolution in (4.10). As discussed in section 4.2.3, taking an L-point DFT is insufficient to capture the correct values for z_l.

4.3 The One-Dimensional Fast Fourier Transform

The Fast-Fourier Transform (FFT) is an algorithm that reduces the complexity of calculating the DFT. As can be seen from the definition of the DFT, (4.1), calculating the DFT of an N-point sequence takes $\mathcal{O}(N^2)$ operations. However, the FFT reduces the algorithmic complexity of the DFT to $\mathcal{O}(N \log N)$. This is a significant increase. In addition, due to the convolution theorem (4.6), we can now calculate circular convolutions in $\mathcal{O}(N \log N)$ as opposed to $\mathcal{O}(N^2)$ operations.

This section will discuss the details of the FFT algorithm and how it achieves its lower computational complexity. Let $\{x_n\}_{n=0}^{N-1}$ be a sequence and $\mathcal{F}[x]_k$ be its DFT. By definition, we have

$$\mathcal{F}[x]_k = \sum_{n=0}^{n-1} x_n W^{nk} \tag{4.11}$$

Suppose $N = N_1 N_2$ with N_1, $N_2 \in \mathbb{N}$. We can redefine the indices n and k with

$$m = N_1 n_1 + n_1, \qquad m_1, k_1 = 0, \ldots, N_1 - 1 \tag{4.12}$$

$$k = N_2 k_1 + k_2, \qquad k_2, m_2 = 0, \ldots, N_2 - 1\,. \tag{4.13}$$

Substitution of (4.12) and (4.13) into (4.11) gives

$$\mathcal{F}[x]_{N_2 k_1 + k_2} = \sum_{n_1=0}^{N_1-1} W^{N_2 n_1 k_1} W^{n_1 k_2} \sum_{n_2=0}^{N_2-1} x_{N_1 n_2 + n_1} W^{N_1 n_2 k_2}\,. \tag{4.14}$$

We define

$$W_1 := e^{-i2\pi/N_1} = W^{N_2} \quad W_2 = e^{-i2\pi/N_2} = W^{N_1}\,,$$

and rewrite (4.14) as

$$\mathcal{F}[x]_{N_2k_1+k_2} = \sum_{n_1=0}^{N_1-1} W_1^{n_1k_1} W^{n_1k_2} \sum_{n_2=0}^{N_2-1} x_{N_1n_2+n_1} W_2^{n_2k_2} \,. \tag{4.15}$$

This shows that the DFT of size N_1N_2 can be viewed as a three step process. First, we calculate N_1 DFTs which correspond to the N_1 values of n_1. We denote these by

$$\mathcal{F}[x]_{n_1,k_2} = \sum_{n_2=0}^{N_2-1} x_{N_1n_2+n_1} W_2^{n_2k_2} \,. \tag{4.16}$$

Second, $\mathcal{F}[x]_{n_1,k_2}$ is multiplied against the *twiddle factors* $W^{n_1k_2}$. Third, $\mathcal{F}[x]_k$ is then obtained by calculating N_2 DFTs of the N_1 DFTs on the N_2 input sequences of $\mathcal{F}[x]_{n_1,k_2}W^{n_1k_2}$, with

$$\mathcal{F}x_{N_2k_1+k_2} = \sum_{n_1=0}^{N_1-1} \mathcal{F}[x]_{n_1,k_2} W^{n_1k_2} W_1^{n_1k_1} \,. \tag{4.17}$$

If we were to calculate the DFT an $N = N_1N_2$ sequence directly, then we would require $N_1^2N_2^2$ multiplications. However, using the algorithm corresponding to (4.15) and (4.17), then computations are broken down into N_1 DFTs of N_2 terms and N_2 DFTs of N_1 terms plus N_1N_2 multiplications for the twiddle factors. Thus the number of multiplications for calculating the DFT of the N_1N_2 points reduces to

$$N_1N_2^2 + N_2N_1^2 + N_1N_2 = N_1N_2(N_1 + N_2 + 1)$$

number of multiplications which is less than the original $N_1^2N_2^2$ multiplications.

While this does not result in the claimed $\mathcal{O}(N \log N)$ multiplications, the algorithm can be used recursively when N is highly composite. We can decompose the original N-point DFT into DFTs of length N_1 and N_2 which can then be further decomposed such that each can be computed using the FFT algorithm. This feature

motivates the application of the FFT to DFT lengths which are the power of an integer, and in most cases N is chosen to be a power of 2 [13].

We can now consider the DFT of an N-point sequence $\{x_n\}_{n=0}^{N-1}$ with $N = 2^t$. In this case, we can do the first stage of the FFT using $N_1 = 2$ and $N_2 = 2^{t-1}$. This results in splitting the original input sequence x_n into two $(N/2)$-point sequences, x_{2n} and x_{2n+1} which correspond to the even and odd indices of the original x_n respectively.

Chapter 5

Discretization of the Collision Integral and Fast Evaluation of Discrete Convolution

5.1 Formulas for Computing the DFT of the Collision Integral Copy and Pasted

To derive the formula for computing the collision operator we rewrite (3.15) as

$$I_{i;j_u,j_v,j_w} = \sum_{i',i''=1}^{s} I_{i,i',i'';j_u,j_v,j_w} \,, \tag{5.1}$$

where

$$I_{i,i',i'';j_u,j_v,j_w} = \sum_{j'_u,j'_v,j'_w=0}^{M-1} \sum_{j''_u,j''_v,j''_w=0}^{M-1} f_{i';j'_u-j_u,j'_v-j_v,j'_w-j_w} f_{i'';j''_u-j_u,j''_v-j_v,j''_w-j_w} A_{i',i'';j'_u,j'_v,j'_w,j''_u,j''_v,j''_w;i} \,.$$

In view of (5.1), we can focus on evaluation of $I_{i,i',i'';j_u,j_v,j_w}$. To simplify the notations in the discussion below, we drop the i, i', and i'' subscripts from $I_{i,i',i'';j_u,j_v,j_w}$, $f_{i';j'_u,j'_v,j'_w}$, $A_{i',i'';j'_u,j'_v,j'_w,j''_u,j''_v,j''_w;i}$, and $\mathcal{F}[I_{i,i',i''}]_{k_u,k_v,k_w}$ and write I_{j_u,j_v,j_w}, $f_{j'_u,j'_v,j'_w}$, $A_{j'_u,j'_v,j'_w,j''_u,j''_v,j''_w}$, and $\mathcal{F}[I]_{k_u,k_v,k_w}$, respectively. In particular, we have

$$I_{j_u,j_v,j_w} = \sum_{j'_u,j'_v,j'_w=0}^{M-1} \sum_{j''_u,j''_v,j''_w=0}^{M-1} f_{j'_u-j_u,j'_v-j_v,j'_w-j_w} f_{j''_u-j_u,j''_v-j_v,j''_w-j_w} A_{j'_u,j'_v,j'_w,j''_u,j''_v,j''_w} \,. \tag{5.2}$$

As is seen from definition (**??**), the multi-dimensional DFT results from applying the one-dimensional DFT along each dimension of the sequence for every fixed value of indices in the other dimensions (see e.g., [13]).

We fix indices j_v and j_w in equation (5.2) and apply the one-dimensional DFT in

the remaining index j_u. Using linearity of the DFT and reordering the sums, we have

$$\mathcal{F}[I_{j_v,j_w}]_{k_u} = \sum_{j'_v,j'_w=0}^{M-1} \sum_{j''_v,j''_w=0}^{M-1} \mathcal{F}[\hat{I}_{j_v,j'_v,j''_v,j_w,j'_w,j''_w}]_{k_u} \,,$$

where

$$\mathcal{F}[\hat{I}_{j_v,j'_v,j''_v,j_w,j'_w,j''_w}]_{k_u} = \sum_{j_u=0}^{M-1} W^{k_u j_u} \hat{I}_{j_u;j_v,j'_v,j''_v,j_w,j'_w,j''_w}$$

$$\hat{I}_{j_u;j_v,j'_v,j''_v,j_w,j'_w,j''_w} = \sum_{j''_u=0}^{M-1} \sum_{j'_u=0}^{M-1} f_{j'_u-j_u,j'_v-j_v,j'_w-j_w} f_{j''_u-j_u,j''_v-j_v,j''_w-j_w} A_{j'_u,j'_v,j'_w,j''_u,j''_v,j''_w} \,.$$

Once again, for purposes of calculating the one-dimensional discrete Fourier transform along dimension j_u, we need only consider the transform of $\hat{I}_{j_u,j_v,j'_v,j''_v,j_w,j'_w,j''_w}$. By similar argument as before, we fix and drop indices j_v, j'_v, j''_v, j_w, j'_w, and j''_w in the latter formula and write

$$\hat{I}_{j_u} = \sum_{j''_u=0}^{M-1} \sum_{j'_u=0}^{M-1} f_{j'_u-j_u} f_{j''_u-j_u,} A_{j'_u,j''_u} \tag{5.3}$$

$$\mathcal{F}[\hat{I}]_{k_u} = \sum_{j_u=0}^{M-1} \sum_{j''_u=0}^{M-1} \sum_{j'_u=0}^{M-1} W^{k_u j_u} f_{j'_u-j_u} f_{j''_u-j_u,} A_{j'_u,j''_u} \,. \tag{5.4}$$

We note that evaluating $\mathcal{F}[\hat{I}]_{k_u}$ directly would require $O(M^3)$ operations. However, taking into consideration the discussion in the last section, expression in the right side of (5.3) can be considered as a circular convolution, having a form similar to (**??**). This motivates us to explore properties of the DFT and rewrite (5.4) in a form suitable for numerical computation. This is accomplished in the following lemma.

Lemma 4 *Let $\{f_j\}_{j=0}^{M-1}$ be a M periodic sequence and $\{A_{ij}\}_{ij}$ be a two index sequence that is M periodic in both its indices. Let $\{\hat{I}_j\}_{j=0}^{M-1}$ be a new sequence defined by*

$$\hat{I}_j = \sum_{j'=0}^{M-1} \sum_{j''=0}^{M-1} f_{j'-j} f_{j''-j} A_{j',j''} \tag{5.5}$$

Let $\mathcal{F}[\hat{I}]_k$ be the DFT of $\hat{I}_j$, then

$$\mathcal{F}[\hat{I}]_k = M \sum_{l=0}^{M-1} \mathcal{F}^{-1}[f]_{k-l} \mathcal{F}^{-1}[f]_l \mathcal{F}[A]_{k-l,l} \tag{5.6}$$

Proof. Applying the one-dimensional DFT to $\hat{I}_j$, we have

$$\mathcal{F}[\hat{I}]_k = \sum_{j_u=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{j''=0}^{M-1} W^{kj} f_{j'-j} f_{j''-j} A_{j',j''} \tag{5.7}$$

We define $\mathcal{F}[A_{j'}]_l$ to be the one-dimensional DFT of $A_{j',j''}$ in the second index, i.e.,

$$\mathcal{F}[A_{j'}]_l = \sum_{j''=0}^{M-1} W^{j''l} A_{j',j''} \,,$$

and rewrite $A_{j',j''}$ as

$$A_{j',j''} = \frac{1}{M} \sum_{l=0}^{M-1} W^{-j''l} \mathcal{F}[A_{j'}]_l \,. \tag{5.8}$$

Substituting (5.8) into (5.7), we have

$$\begin{aligned} \mathcal{F}[\hat{I}]_k &= \frac{1}{M} \sum_{j=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{j''=0}^{M-1} W^{jk} f_{j'-j} f_{j''-j} \left(\sum_{l=0}^{N-1} W^{-j''l} \mathcal{F}[A_{j'}]_l \right) \\ &= \frac{1}{M} \sum_{l=0}^{M-1} \sum_{j=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{j''=0}^{M-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l \end{aligned} \tag{5.9}$$

Consider the sum that runs over index j''. Assuming that indices l, j, and j' are held constant, we split the sum into two parts.

$$\begin{aligned} &\sum_{j''=0}^{M-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l \\ &= \sum_{j''=0}^{j-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l + \sum_{j''=j}^{M-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l \,. \end{aligned} \tag{5.10}$$

Notice $j'' - j < 0$ in the first sum. Using the periodicity of f_j the summation index can be redefined so that only values $f_{j''-j}$ with positive $j'' - j$ appear in the sum. Indeed, we assume that $j'' < j$ and notice that $f_{j''-j} = f_{j''-j+M}$ since f_j is M periodic. We also have $W^M = 1$, so $W^{j''l} = W^{j''l+Ml} = W^{(j''+M)l}$. Introducing $\hat{j}'' = j'' + M$, we observe

$$\begin{aligned}\sum_{j''=0}^{j-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l &= \sum_{j''=0}^{j-1} W^{jk} f_{j'-j} f_{j''-j+M} W^{(-j''+M)l} \mathcal{F}[A_{j'}]_l \\ &= \sum_{\hat{j}''=M}^{M-1+j} W^{jk} f_{j'-j} f_{\hat{j}''-j} W^{-\hat{j}''l} \mathcal{F}[A_{j'}]_l \,.\end{aligned}$$

Combining the last formula with (5.10) we have

$$\sum_{j''=0}^{M-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l = \sum_{j''=j}^{M-1+j} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l \,. \tag{5.11}$$

Introducing a substitution of index $u'' = j'' - j$ we rewrite the right side of (5.11) as follows

$$\sum_{j''=0}^{M-1} W^{jk} f_{j'-j} f_{j''-j} W^{-j''l} \mathcal{F}[A_{j'}]_l = \sum_{u''=0}^{M-1} W^{jk} f_{j'-j} f_{u''} W^{(-u''-j)l} \mathcal{F}[A_{j'}]_l \,.$$

Going back to (5.9), we replace the inside sum with the last expression to have

$$\begin{aligned}&\frac{1}{M} \sum_{l=0}^{M-1} \sum_{j=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{u''=0}^{M-1} W^{jk} f_{j'-j} f_{u''} W^{(-u''-j)l} \mathcal{F}[A_{j'}]_l \\ &\qquad = \frac{1}{M} \sum_{l=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{u''=0}^{M-1} W^{-u''l} f_{u''} \left(\sum_{j=0}^{M-1} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l \right) .\end{aligned} \tag{5.12}$$

Now we focus on the term within the parentheses in (5.12). Splitting the sum and

using periodicity, we obtain

$$\sum_{j=0}^{M-1} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l \tag{5.13}$$

$$= \sum_{j=0}^{j'} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l + \sum_{j=j'+1}^{M-1} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l$$

$$= \sum_{j=0}^{j'} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l + \sum_{j=j'+1}^{M-1} W^{(j-M)(k-l)} f_{j'-j+M} \mathcal{F}[A_{j'}]_l$$

$$= \sum_{j=0}^{j'} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l + \sum_{\hat{j}=j'-M+1}^{-1} W^{\hat{j}(k-l)} f_{j'-\hat{j}} \mathcal{F}[A_{j'}]_l$$

$$= \sum_{j=j'-M+1}^{j'} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l = \sum_{u'=0}^{M-1} W^{(j'-u')(k-l)} f_{u'} \mathcal{F}[A_{j'}]_l \,. \tag{5.14}$$

Here $u' = j' - j$. Substituting this result into (5.12) and regrouping sums, we yield

$$\frac{1}{M} \sum_{l=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{u''=0}^{M-1} W^{-u''l} f_{u''} \left(\sum_{j=0}^{M-1} W^{j(k-l)} f_{j'-j} \mathcal{F}[A_{j'}]_l \right)$$

$$= \frac{1}{M} \sum_{l=0}^{M-1} \sum_{j'=0}^{M-1} \sum_{u''=0}^{M-1} W^{-u''l} f_{u''} \left(\sum_{u'=0}^{M-1} W^{(j'-u')(k-l)} f_{u'} \mathcal{F}[A_{j'}]_l \right)$$

$$= M \sum_{l=0}^{M-1} \left(\frac{1}{M} \sum_{u'=0}^{M-1} W^{-u'(k-l)} f_{u'} \right) \left(\frac{1}{M} \sum_{u''=0}^{M-1} W^{-u''l} f_{u''} \right) \left(\sum_{j'=0}^{M-1} W^{j'(k-l)} \mathcal{F}[A_{j'}]_l \right). \tag{5.15}$$

The terms in the parentheses in (5.15) are just the definitions of the DFT. Thus we can write the equation as

$$\mathcal{F}[\hat{I}]_k = M \sum_{l=0}^{M-1} \mathcal{F}^{-1}[f]_{k-l} \mathcal{F}^{-1}[f]_l \mathcal{F}[A]_{k-l,l} \,.$$

■

Lemma 4 allows us to compute (5.4) in $O(M^2)$ operations. Indeed, it takes $O(M \log M)$ operations to compute $\mathcal{F}^{-1}[f]_{k_u}$ using a fast Fourier transform and it takes $O(M^2)$ operations to compute discrete convolution in the frequency space (5.6). To extend this result to $\mathcal{F}[I]_{k_u,k_v,k_w}$, it is sufficient to repeat the approach for indices j_v and j_w focusing on one dimension at a time. The following theorem summarizes the result.

Theorem 2 *Let f_{j_u,j_v,j_w} be a three-index sequence that is periodic in each index with period M and let $A_{j'_u,j'_v,j'_w,j''_u,j''_v,j''_w}$ be a M-periodic six-dimensional tensor. The multi-dimensional discrete Fourier transform of equation (5.2) can be represented as*

$$\mathcal{F}[I]_{k_u,k_v,k_w} = M^3 \sum_{l_u,l_v,l_w=0}^{M-1} \mathcal{F}^{-1}[f]_{k_u-l_u,k_v-l_v,k_w-l_w} \mathcal{F}^{-1}[f]_{l_u,l_v,l_w} \mathcal{F}[A]_{k_u-l_u,k_v-l_v,k_w-l_w,l_u,l_w,l_w} \tag{5.16}$$

Proof. We apply the one dimensional discrete Fourier transform along j_u in equation (5.2) and apply Lemma 4:

$$\begin{aligned}
\mathcal{F}[I_{j_v,j_w}]_{k_u} &= \sum_{j'_v,j'_w=0}^{M-1} \sum_{j''_v,j''_w=0}^{M-1} \left(\sum_{j_u=0}^{M-1} \sum_{j'_u=0}^{M-1} \sum_{j''_u=0}^{M-1} W^{j_u k} f_{j'_u-j_u,j'_v-j_v,j'_w-j_w} f_{j''_u-j_u,j''_v-j_v,j''_w-j_w} A_{j'_u,j'_v,j'_w,j''_u,j''_v,j''_w} \right) \\
&= M \sum_{j'_v,j'_w=0}^{M-1} \sum_{j''_v,j''_w=0}^{M-1} \sum_{l_u=0}^{M-1} \mathcal{F}^{-1}[f_{j'_v-j_v,j'_w-j_w}]_{k_u-l_u} \mathcal{F}^{-1}[f_{j''_v-j_v,j''_w-j_w}]_{l_u} \mathcal{F}[A_{j'_v,j'_w,j''_v,j''_w}]_{k_u-l_u,l_u} \\
&= M \sum_{l_u=0}^{M-1} \sum_{j'_w=0}^{M-1} \sum_{j''_w=0}^{M-1} \\
&\qquad \left(\sum_{j'_v=0}^{M-1} \sum_{j''_v=0}^{M-1} \mathcal{F}^{-1}[f_{j'_v-j_v,j'_w-j_w}]_{k_u-l_u} \mathcal{F}^{-1}[f_{j''_v-j_v,j''_w-j_w}]_{l_u} \mathcal{F}[A_{j'_v,j'_w,j''_v,j''_w}]_{k_u-l_u,l_u} \right).
\end{aligned} \tag{5.17}$$

We now focus on the terms inside the parentheses. We fix the indices j_w, j'_w, j''_w,

k_u, and l_u in the grouped terms We drop these indices and write

$$\tilde{I}_{j_v} = \sum_{j'_v=0}^{M-1} \sum_{j''_v=0}^{M-1} \tilde{f}_{j'_v - j_v} \tilde{f}_{j''_v - j_v} \tilde{A}_{j'_v, j''_v} ,$$

where

$$\tilde{f}_{j'_v - j_v} = \mathcal{F}^{-1}[f_{j'_v - j_v}], \qquad \tilde{A}_{j'_v, j''_v} = \mathcal{F}[A_{j'_v, j''_v}].$$

We can see that this expression is identical to (5.5). We take the discrete Fourier transform along the j_v index of $\mathcal{F}[\tilde{I}]_{k_v}$ and apply Lemma 4 to arrive at

$$\mathcal{F}[\tilde{I}]_{k_v} = M \sum_{l_v=0}^{M-1} \mathcal{F}^{-1}[\tilde{f}]_{k_v - l_v} \mathcal{F}^{-1}[\tilde{f}]_{l_v} \mathcal{F}[\tilde{A}]_{k_v - l_v, l_v} . \tag{5.18}$$

We recall definitions of $\tilde{f}_{j'_v - j_v}$ and $\tilde{A}_{j'_v, j''_v}$ and notice that the multi-index Fourier transform results from applying the one-dimensional transform in each index. Bringing indices j'_w, j''_w, k_u and l_u back, equation (5.18) becomes

$$\mathcal{F}[I_{j_w}]_{k_u, k_v} = M^2 \sum_{l_u, l_v=0}^{M-1} \mathcal{F}^{-1}[f_{j'_w - j_w}]_{k_u - l_u, k_v - l_v} \mathcal{F}^{-1}[f_{j''_w - j_w}]_{l_u, l_v} \mathcal{F}[A_{j'_w, j''_w}]_{k_u - l_u, k_w - l_w, l_u, l_w} .$$

Performing the discrete Fourier transform in the j_w and repeating the argument once more we arrive at the statement of the theorem.

■

5.2 The Algorithm and its Complexity Copy and Pasted

Theorem 2 allows us to calculate the collision operator (3.15) in $O(s^3 M^6)$ operations using the algorithm outlined below. We note that $\mathcal{F}[A_{i,i',i''}]_{k_u, k_v, k_w, l_u, l_w, l_w}$ can be precomputed and therefore does not factor into the algorithmic complexity analysis.

1. The first step of the algorithm is to evaluate $\mathcal{F}^{-1}[f_i]_{k_u, k_v, k_w}$. Evaluation of the

inverse Fourier transform requires $O(M^3 \log M)$ operations for each value of index i by utilizing three-dimensional FFT. This must be repeated for each i, resulting in the total of $O(s^3 M^3 \log M)$ operations where s^3 is the number of velocity nodes in each velocity cell.

2. Next we directly compute the convolution

$$\mathcal{F}[I_{i,i',i''}]_{k_u,k_v,k_w} = M^3 \sum_{l_u,l_v,l_w=0}^{M-1} \mathcal{F}^{-1}[f_{i'}]_{k_u-l_u,k_v-l_v,k_w-l_w} \mathcal{F}^{-1}[f_{i''}]_{l_u,l_v,l_w} \mathcal{F}[A_{i,i',i''}]_{k_u-l_u,k_v-l_v,k_w-l_w,l_u,l}$$

using periodicity of both $\mathcal{F}^{-1}[f_i]_{l_u,l_v,l_w}$ and $\mathcal{F}[A_{i,i',i''}]_{k_u,k_v,k_w,l_u,l_w,l_w}$.

For fixed values of indices i, i', i'' and k_u, k_v, k_w, calculating $\mathcal{F}[I_{i,i',i''}]_{k_u,k_v,k_w}$ requires $O(M^3)$ arithmetic operations. There are M^3 combinations of k_u, k_v, k_w and s^3 combinations of indices i, i', and i'', therefore complexity of this step is $O(s^3 M^6)$.

3. Linearity of the Fourier transform allows us to sum $\mathcal{F}[I_{i,i',i''}]_{k_u,k_v,k_w}$ along i', i'' to calculate $\mathcal{F}[I_i]_{k_u,k_v,k_w}$.

$$\mathcal{F}[I_i]_{k_u,k_v,k_w} = \sum_{i',i''=1}^{s} \mathcal{F}[I_{i,i',i''}]_{k_u,k_v,k_w} \,.$$

This step requires adding s^2 sequences of length M^3 for every value of i, resulting in a complexity of $O(s^3 M^3)$ operations.

4. We recover $\mathcal{F}^{-1}[\mathcal{F}[I_i]]_{j_u,j_v,j_w} = I'_{i;j_u,j_v,j_w}$. This requires calculating the three-dimensional inverse DFT for every i which gives a complexity of $O(sM^3 \log M)$.

Overall, the algorithm has the numerical complexity of $O(s^3 M^6)$ dominated by step 2. We note that $s = s_u s_v s_w$ is usually kept fixed and the number of cells M^3 in velocity domain is changing. In this case, the main contribution to complexity growth comes from M, the number of velocity cells in one velocity dimension. Thus

we can consider the algorithm to be of complexity $O(M^6)$. In our simulations, we kept $s_u, s_v, s_w \leq 3$, however higher values may be used too, at least theoretically. The results of this analysis are validated within the next chapter.

In the implementation of the algorithm, we use the tensor product ordering for f. However, after taking the inverse FFT of $\mathcal{F}[I_i]_{i;j_u,j_v,j_w}$, we get differently ordered sequence, $I'_{i;j_u,j_v,j_w}$. This is due to the fact we take the FFT with respect to j, the shift in velocity cell. To retrieve back $I_{i;j_u,j_v,j_w}$ from $I'_{i;j_u,j_v,j_w}$, the following equation was used:

$$j_u = (j'_u - j_{u;c} \mod M)\,,\ j_v = (j'_v - j_{v;c} \mod M)\,,\ j_w = (j'_w - j_{w;c} \mod M)\,,$$

where $j_u u; c$ is canonical node, j'_u is the old index in $I_{i;j_u,j_v,j_w}$, and the mod operation always returns the positive remainder. To be explicit, the algorithm below

Algorithm 1 Perform FFT convolution at a single point in space.

function FFTBOLTZMANN(f,$\mathcal{F}[A]$)
  Input f, Q ▷ The solution at time t.
  Input $\mathcal{F}[A]$ ▷ The DFT of the A operator.
  for i=1,s **do**
    $F_i = \mathcal{F}^{-1}[f]$ ▷ Calculate the inverse DFT of f along velocity cells.
  end for
  for Each combination of $i, i', i'' = 1, s$ **do**
    $\mathcal{F}[I_{i,i',i''}] = M^3 \sum_{l_u,l_v,l_w=0}^{M-1} \mathcal{F}^{-1}[f_{i'}]_{k_u-l_u,k_v-l_v,k_w-l_w} \mathcal{F}^{-1}[f_{i''}]_{l_u,l_v,l_w} \mathcal{F}[A_{i,i',i''}]_{k_u-l_u,k_v-l_v,k_w-l_w,l_u,}$
    $\mathcal{F}[I_i] = \mathcal{F}[I_i] + \mathcal{F}[I_{i,i',i''}]$
  end for
  $I_i' = \mathcal{F}^{-1}[\mathcal{F}[I_i]]$ ▷ Take inverse DFT of $\mathcal{F}I_i$ along velocity cells.
  for i=1,s **do** ▷ Reordering step.
    for j=1,M **do**
      $j_u' = j_u - j_{u;c} \mod M$
      $j_v' = j_u - j_{v;c} \mod M$
      $j_w' = j_w - j_{w;c} \mod M$
      $I_{j_u,j_v,j_w;i} = I_{j_u',j_v',j_w';i}$
    end for
  end for
  return I_i
end function

5.3 Periodic Continuation of f and A

Theorem 2 required that we take both a periodic continuation of f and A in order to calculate $\mathcal{F}[i]$. However, this assumption does not make *physical* sense as it would imply there was infinite energy within the system. This means that we must periodically extend $f(t, \vec{x}, \vec{v})$ outside the domain in the $\vec{v}$ direction and $A(\vec{v}, \vec{v}_1; \phi_{i;c})$ in both $\vec{v}$ and $\vec{v}_1$. However, this assumption causes overlap when calculating the circular convolutions. Since $f_j > 0$, we expect that circular and linear convolutions to give different results. For example, when calculating (3.15), we can no longer set $f_j = 0$ when $j \leq 0$ or $j > M$. We assume that the velocity domain is sufficiently large so that the support of both f and A are limited to half of the domain. This would be equivalent to zero-padding the sequence. The discussion in section 4.2.3 demonstrated that it is possible to zero-pad the sequences in order restore equivalence between the

circular and linear convolutions. We can zero-pad (3.15) which requires zero-padding to length $2N-2$ in each dimension. This requires storing $(N-1)^6$ zeroes for $A_{j',j''}$. This causes the memory requirements to grow rapidly, so this approach can be used, but it must be used with caution or memory requirements will be exceeded.

However, for most calculations, we take a different approach. The effects of periodic extension and truncation of $f(t,\vec{x},\vec{v})$ on the collision operator were considered in [14]. However, the effects of periodic continuation and truncation of $A(\vec{v},\vec{v}_1;\phi_{i;c})$ are much less understood. We can see from (3.8) that the kernle is growing linearly towards infinty in the direction of $\vec{v}-\vec{v}_1$ for at least some points of $\vec{v}$. A truncation of kernel $A(\vec{v},\vec{v}_1;\phi_{i;c})$ was used in [2] in which entries of $A(\vec{v},\vec{v}_1;\phi_{i;c})$ were set equal to zero if $\|\vec{v}-\vec{v}_1\|<R$ for some $R>0$. Numerical simulations in [2] confirmed that the effects of truncation are negligible if the support of the solution can be enclosed in a ball of diameter R. Numerical simulations will demonstrate that the direct convolution in (3.15) can be treated as a circular convolution so long as the support of the solution is sufficiently small.

In addition, certain properties of f lend themselves to more accurate calculations. Typically, in the boundaries of the discretization of f, the values of f are close to zero in the sense that the values are typically a few orders of magnitude smaller than the bulk-velocity since the shape due to nature of the distribution of gasses. In essence, the closer to Maxwellian the distribution is, the more the tails of the distribution diminish. This allows us to take the circular convolution while adding minimal error.

Using this idea, we can still achieve good accuracy while using minimal zero-padding. We sometimes need only to pad by a small number of zeroes (generally ≤ 4) to lose any effects of aliasing. This is shown in later numerical results.

Chapter 6

Numerical Results

This chapter will cover the numerical results of computing the collision operator using the nodal-DG velocity discretizations and convolution using the DFT.

6.1 Reduction in Computational Complexity

This section will cover estimating the numerical complexity of the method. The analysis in section 5.2 suggested that the algorithm had a total number of operations of $O(M^6)$ where M is the number of velocity cells in one velocity dimension. As discussed in [2], the direction evaluation of convolution (3.15) requires $O(M^8)$ operations. Thus we expect the FFT evaluation of (3.15) to take much less time than the method discussed in [2].

The results in Table 6.1 display the CPU times for evaluating the collision operator at one spatial point using both the FFT and direct evaluations. The computations were done on a Intel Core i7-3770 3.4 GHz processor. The code was compiled using the Intel Fortran Compilers along with Intel Math Kernal Library. The number of cells in the velocity domain were varied from 9 to 27. $s = 1$ in each run. The computational complexity was modeled using $t = O(M^\alpha)$ where α is a constant. The estimated values of α were calculated as $\hat{\alpha} = \ln(M_1/M_2)/\ln(t_1/t_2)$.

Note that in the case of the Fourier evaluation, the observed orders are significantly higher than the projected value of 6. Still, the orders are significantly lower than the orders of the direct evaluation. Deviations from the theoretical estimate of $\alpha = 6$ may be due to the costs of the memory transfer operations and due to the choice of the specific fast Fourier transform that was automatically selected by the KML library based on the value of M. Overall, the new approach showed a dramatic improvement

in speed as compared to the direct evaluation of the collision operator used in [1]. The acceleration is expected to be even larger for higher values of M.

	FFT		Direct		Speedup
M	time, s	$\hat{\alpha}$	time, s	$\hat{\alpha}$	
9	1.47E-02		1.25E-01		8.5
15	3.94E-01	6.43	4.91E+00	7.18	12.5
21	3.09E+00	6.14	7.80E+01	8.21	25.2
27	1.64E+01	6.65	6.05E+02	8.15	36.7

Table 6.1: CPU times for evaluating the collision operator directly and using the Fourier transform.

6.2 Numerical Results of the Split Form of the Operator

As was discussed in section 3.3, there is an alternative formulation of (3.7) which we write as (3.9). This section discusses the numerical issues with conservation that this form had.

In Figure 6.1 results of the evaluation of the collision integral at a single spatial point are presented for both split and non-split formulations. The value of the solution $f(t, \vec{v})$ in these computations is given by the sum of two Maxwellian distributions with dimensionless densities, bulk velocities, and temperatures given as follows: $n_1 = 1.6094$, $n_2 = 2.8628$, $\bar{\vec{u}}_1 = (0.7750, 0, 0)$, $\bar{\vec{u}}_2 = (0.4357, 0, 0)$, $T_1 = 0.3$, and $T_2 = 0.464$. These values correspond to upstream and downstream conditions of a normal shock wave with the Mach number 1.55. Discretization of the solution was done using 27 velocity cells in each dimension and one velocity node per cell. The collision operator was evaluated using both the split and non-split forms and using both direct evaluation and evaluation using the Fourier transform. Results of the direct evaluation of the split form of the collision operator are shown in plots (b) and (e). Notably, values of the collision operator are zero at the boundary of the domain, which is what one would expect from the collision process. Results of evaluation of the split form of the collision operator using the Fourier transform are shown in plots (a) and (b).

Significant non-zero values can be observed at the corners of the domain. This is likely to be a manifestation of aliasing. Results of evaluating the non-split form of the collision operator using the Fourier transform are shown in plots (c) and (f). One can notice that in the case of the non-split form, aliasing is not visible. In fact, the L^1-norm of the difference between the direct and Fourier evaluations of the non-split collision operator in this case was 2.9E-4 and the L^∞-norm was 1.1E-4. We note that the diameters of the support of the collision kernels are comparable in both split and non-split cases. However, the non-split kernel is more sparse. We also note that by padding the solution and the collision kernel with zeros, the aliasing errors can be reduced in the case of the split formulations. However, this will also increase memory and time costs of calculations. The non-split form of the collision operator has significantly smaller aliasing errors and does not require zero padding. Therefore, it is more efficient.

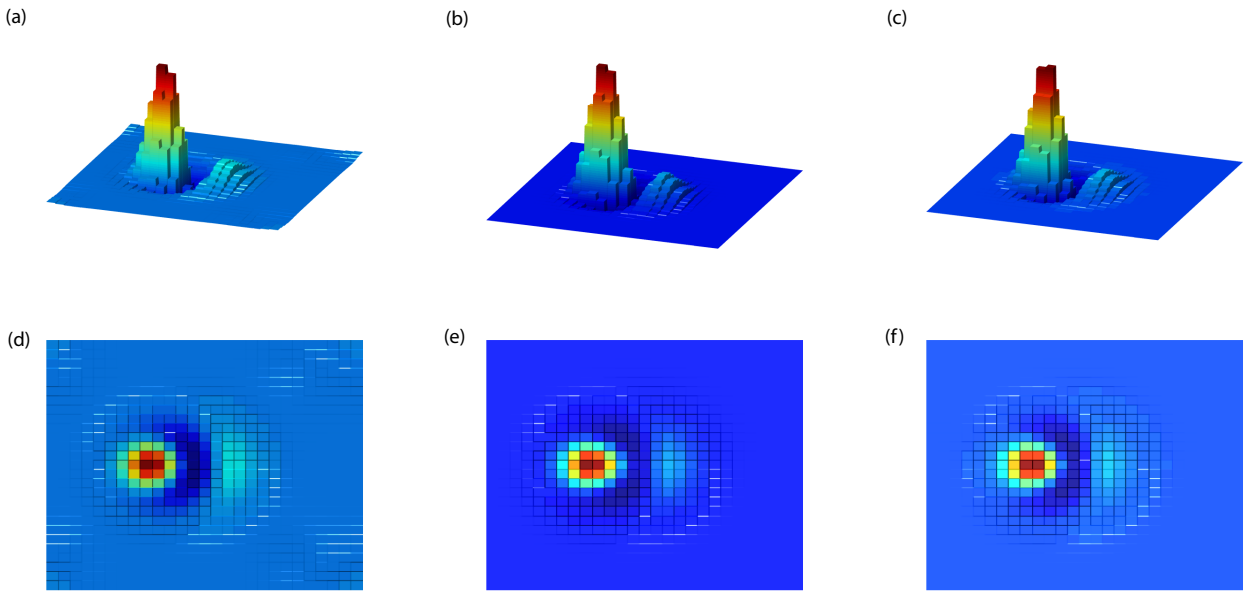

Figure 6.1: Evaluation of the collision operator using split and non-split forms: (a) and (d) the split form evaluated using the Fourier transform; (b) and (e) the split form evaluated directly; (c) and (f) the non-split form evaluated using the Fourier transform.

Another important issue that makes the non-split formulation more attractive is concerned with conservation of mass, momentum, and energy in the discrete solutions. It is the property of the exact Boltzmann collision operator that its mass, momentum, and temperature moments are zero. Generally, the conservation laws are satisfied only approximately when the Boltzmann equation is discretized. Many numerical approaches include mechanisms dedicated to enforcement of the conservation laws in discrete solutions in order to guarantee a physically meaningful result.

	Error in Conservation of Mass				Error in Conservation of Temperature			
	Split		Non-split		Split		Non-split	
n	Fourier	Direct	Fourier	Direct	Fourier	Direct	Fourier	Direct
9	0.37	1.26	1.71E-5	1.92E-5	3.51	1.69	1.71E-2	1.84E-2
15	0.10	1.20	1.45E-5	1.71E-5	0.29	1.25	1.64E-3	3.15E-3
21	0.18	1.18	0.67E-5	0.93E-5	1.38	1.24	5.61E-5	1.75E-3
27	0.18	1.18	0.61E-5	0.86E-5	1.37	1.24	5.40E-4	1.05E-3

Table 6.2: Absolute errors in conservation of mass and temperature in the discrete collision integral computed using split and non-split formulations.

It was observed that if no measures are introduced to enforce the conservation laws, solutions to the problem of spatially homogeneous relaxation obtained using the split formulation of the collision integral exhibit large, on the order of 5% errors in temperature. The mass and momentum are also poorly conserved in this case. At the same time, solutions obtained using the non-split formulation had their mass, momentum, and temperature accurate to three or more digits. To further explore this phenomena, we evaluated the collision operator in both split and non-split forms and computed its mass, momentum, and temperature moments. The solution was taken to be the sum of two Maxwellians in the example above. The numbers of velocity cells were varied from 9 to 27. In both split and non-split formulations of the collision integral, the decomposed form (3.17) of the solution was used. For both

forms, evaluation of the collision operator was done directly and using the Fourier transform. The results are summarized in Table 6.2. It can be seen that errors in the mass and temperature in the non-split formulation are several orders of magnitude smaller than in the split formulation. The errors are also larger in the case of direct evaluation. A possible explanation to this is the combined effect of finite precision arithmetic and truncation errors in integration that lead to catastrophic cancellation when gain and loss terms are combined. We note that in both split and non-split forms, fulfilment of conservation laws requires exact cancellation of the respective integration sums. When the gain and loss terms are computed separately using numerical quadratures, the relative truncation errors are expected to be acceptable for each of the terms. This may change, however, when the terms are combined. It is conceivable that significant digits cancel in the two terms and the truncation errors are promoted into significance, manifesting in strong violations of conservation laws. At the same time, increasing the number of velocity cells may not remedy the problem due to the expected accumulation of roundoff errors. Indeed, evaluation of the gain term in (3.9) requires $O(M^8)$ arithmetic operations. It is possible that combination of large and small values in the finite precision arithmetic results in loss of low order digits and a significant accumulation of roundoff. When the gain and loss terms are combined, this, again, will lead to loss of significance and to perturbations of conservation laws. In the case when both the non-split form and the decomposition (3.17) are used, much of the cancellation is happening on the level of the integrand. We hypothesize here that the resulting values of the integrand are smaller and vary less in scale. As a result, the accumulated absolute truncation and roundoff errors are also smaller, which gives better accuracy in conservation laws.

Because of the poor conservation properties and because of the susceptibility to aliasing errors we do not recommend the split form (3.9) for numerical implementation.

6.3 0d Homogeneuous Relaxation

In this section we present results of solution of the problem of spatially homogeneous relaxation using Fourier evaluation of the collision operator. Two cases of initial data were considered. In both cases, the initial data is a sum of two Maxwellian densities. In the first case, the dimensionless densities, bulk velocities, and temperatures of the Maxwellians are $n_1 = 1.0007$, $n_2 = 2.9992$, $\bar{\vec{u}}_1 = (1.2247, 0, 0)$, $\bar{\vec{u}}_2 = (0.4082, 0, 0)$, $T_1 = 0.2$, $T_2 = 0.7333$. These parameters correspond to upstream and downstream conditions of the Mach 3 normal shock wave. In the second case, we use the parameters of the example of the previous section: $n_1 = 1.6094$, $n_2 = 2.8628$, $\bar{\vec{u}}_1 = (0.7750, 0, 0)$, $\bar{\vec{u}}_2 = (0.4357, 0, 0)$, $T_1 = 0.3$, and $T_2 = 0.464$. These parameters correspond to upstream and downstream conditions of a Mach 1.55 shock wave.

In Figures 6.2 and 6.4, relaxation of moments in the Mach 3.0 and Mach 1.55 solutions are presented. In the case of Mach 3.0, $M = 33$ velocity cells were used in each velocity dimension with one velocity node on each cell, $s = 1$. In the case of Mach 1.55, $M = 15$ and $s = 1$ were used. In the computed solutions, the collision operator was evaluated both using the Fourier transform and directly. In the Mach 3.0 instance, the directional temperature moments were compared to the moments obtained from a DSMC solution [6].

It can be seen that the solutions obtained by the Fourier evaluation of the collision integral are close to those computed by the direct evaluation. The low order moments are in excellent agreement for both presented solutions. However, there are differences

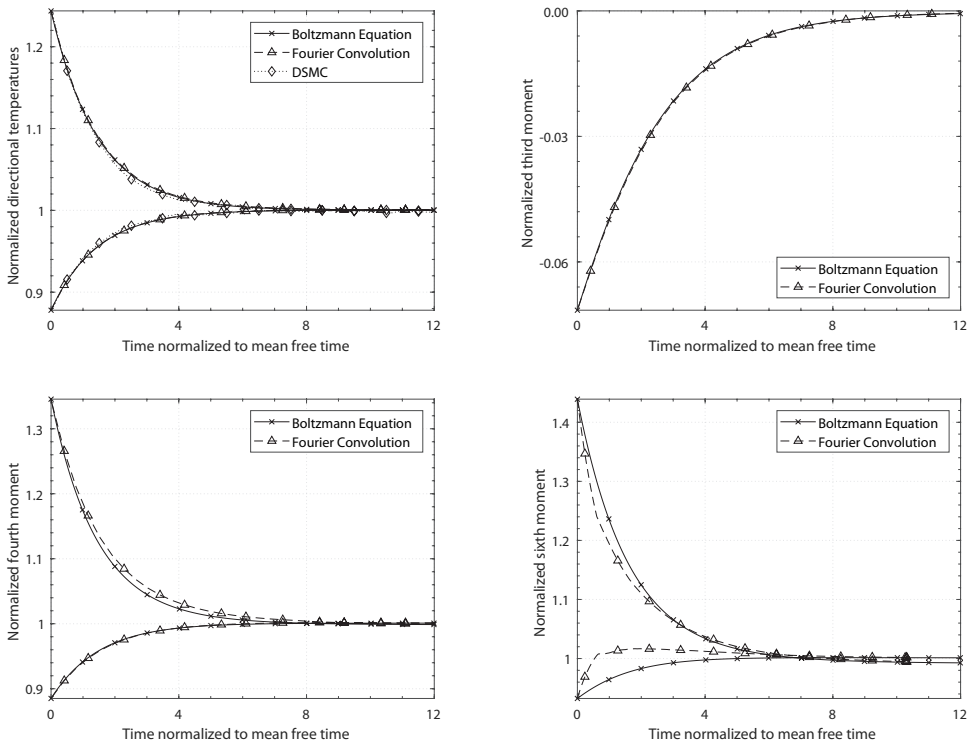

Figure 6.2: Relaxation of moments $f_{\varphi_{i,p}} = \int_{R^3} (u_i - \bar{u}_i)^p f(t, \vec{u})\, du$, $i = 1, 2$, $p = 2, 3, 4, 6$ *in a mix of Maxwellian streams corresponding to a shock wave with Mach number 3.0 obtained by solving the Boltzmann equation using Fourier and direct evaluations of the collision integral. In the case of* $p = 2$*, the relaxation of moments is also compared to moments of a DSMC solution [6].*

in the higher moments. It appears that the differences are caused by a small amount of the aliasing error in the solutions. This can be reduced by padding the solution and the kernel with zeros at the expense of higher numerical costs, both in time and memory. Overall, however, the $O(M^6)$ evaluation of the collision operator using the Fourier transform appears to be consistent and stable.

It can be seen in Figure 6.2 that the higher moments suffer more error than the lower moments. It can be seen that the fourth moment has modest divergences from the direct evaluation, while the sixth moment diverges greatly in the direct evaluation. This author believes that this due to the fact that we have more aliasing towards the ends of the domain. The higher moments will give much more weight to those parts

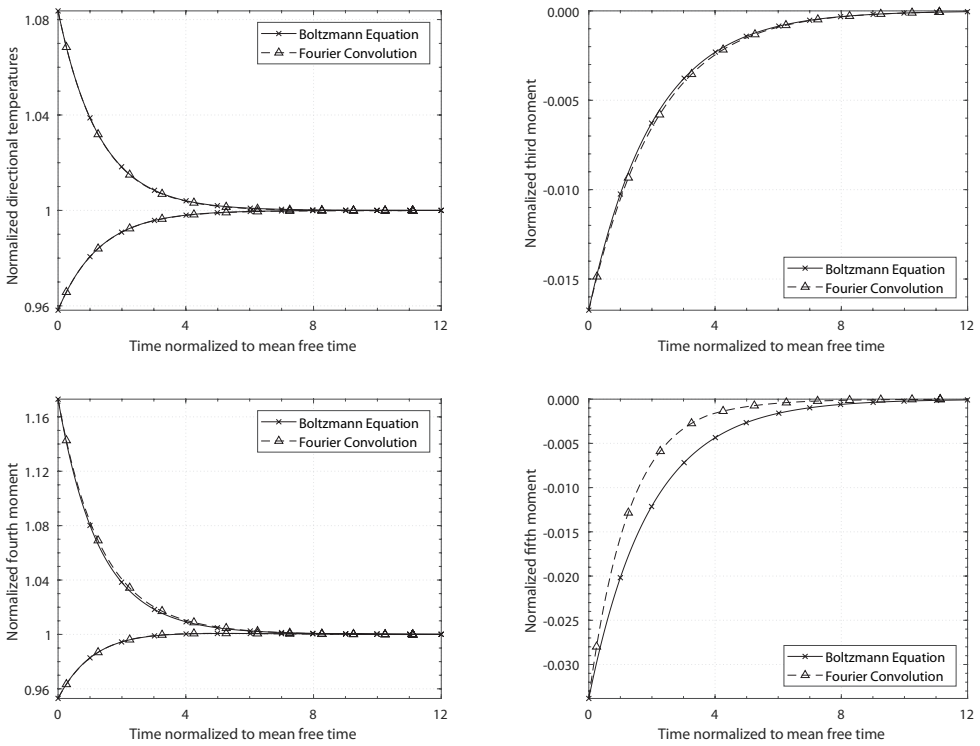

Figure 6.3: Relaxation of moments $f_{\varphi_{i,p}}$, $i = 1, 2$, $p = 2, 3, 4, 6$ in a mix of Maxwellian streams corresponding to a shock wave with Mach number 1.55 obtained by solving the Boltzmann equation using Fourier and direct evaluations of the collision integral.

of the domain than earlier moments. Indeed, Figure 6.1 supports this hypothesis as we can see the edges of the domain had gain mass.

6.4 Zero-Padding

As discussed in section 4.2.2, we saw that it is possible to reduce the amount of error from the calculations if we zero-pad the resulting sequence. These results show how zero-padding affects the error of the method as a function of the number of zeros that are added to the sequence.

In section 4.2.2, we showed that if want to completely eliminate any aliasing effect, then we must pad f to length $2M - 1$ completely remove any aliasing effects from taking the inverse DFT to retrieve back I_j. However, this is prohibitive since that would mean adding $\mathcal{O}(M^6)$ zeros to A in order to remove any aliasing effects from

$\mathcal{F}[A]$. Since we need to double its length in each dimension, that results in a 64 times increase in the memory storage of $\mathcal{F}[A]$. As discussed in section 5.3, we assume that the shape of the function helps with the zero-padding and argue that we need only modest zero-padding to eliminate aliasing error. However, we can combine modest zero padding with the fact that f approaches zero in the edges of the domain to remove the numerical effects of aliasing rather quickly.

Zero-padding was accomplished by adding n_{pad} zeros to each side of the sequence of f. Note that since f is 3-dimensional, this means that symmetrically padding f with n_{pad} zeros on each size results in a size increase of $8n_{\text{pad}}^3$ (where the 8 comes from the fact that we are symmetrically padding) in the amount of numbers that must be stored. In addition, to naively perform the calculations, we must also zero-pad the A kernel. This results in a memory increase of $64n_{\text{pad}}^6$. This of course then becomes restrictive as the number zeros increases since we will then run out of memory rather quickly. However, as the table below shows, even a modest increase in the number of zeros results in much lower error compared to the the straight-forward method presented earlier.

Table 6.3 shows how zero padding affected the accuracy of the method compared to a direct evaluation. These results were gathered using the parameters: $n_1 = 1.6094$, $n_2 = 2.8628$, $\bar{\vec{u}}_1 = (0.7750, 0, 0)$, $\bar{\vec{u}}_2 = (0.4357, 0, 0)$, $T_1 = 0.3$, and $T_2 = 0.464$ which corresponds to a mach 1.55 simulation. The table shows the error between the DFT method and direct method by symmetrically padding by zeros. Note that $n_{\text{pad}} = 1$ means that two 0s were added, one 0 to both ends of the sequence.

The table shows that the zero-padding the method results in more accuracy in the method. Remarkably, very little zero-padding must be done for the method to converge in accuracy. For both $N = 9$ and $N = 15$, the method converges in accuracy

	$N = 9$		$N = 15$		$N = 21$	
n_{pad}	$L_{\max}$	L_1	$L_{\max}$	L_1	$L_{\max}$	L_1
0	4.93E-05	3.74E-04	6.08E-05	4.66E-04	8.51E-05	5.39E-04
1	5.19E-07	1.15E-06	2.61E-07	1.33E-06	1.18E-06	8.79E-06
2	5.19E-07	1.14E-06	2.61E-07	2.93E-07	8.73E-08	9.59E-08
3	5.19E-07	1.14E-06	2.61E-07	2.93E-07	7.33E-09	9.59E-08
4	5.19E-07	1.14E-06	2.61E-07	2.93E-07	8.73E-08	9.58E-08

Table 6.3: The $L_{\max}$ and L_1 errors as we increase n_{pad}.

after symmetrically padding by two zeros. $N = 21$ converged after $n_{\text{pad}} = 3$. These results suggest that the assumption of f approaching zero rapidly in the edges of the domain is a valid one since not many zeros are needed to reach a convergence in accuracy. In addition, the author performed padding up to $n_pad\text{pad} = 8$ for the $N = 15$ case without seeing any substantial change from the above $n_{\text{pad}} = 4$ case (and therefore omitted from the above table). This of course is much beyond the required $M - 2$ zeros needed to achieve perfect accuracy, so any remaining error is an artifact of numerics.

However, padding by zeros will have an impact on the performance of the method as we increase the number of operations required to fully complete the calculations. The table below shows the amount of seconds the DFT along with how long the direct method took without padding.

	$N = 9$		$N = 15$		$N = 21$	
n_{pad}	time (s)	speed up	time (s)	speed up	time (s)	speed up
0	0.22	6.28	1.18	40.38	6.09	114.30
1	0.19	7.27	1.65	28.95	11.18	62.27
2	0.31	4.36	6.23	7.66	33.19	20.97
3	0.97	1.41	6.30	7.58	33.10	21.03
4	2.13	0.64	11.40	4.19	51.07	13.63

Table 6.4: How performance of the method decreases as we increase n_{pad}.

As we can see from the table, the the fourier method still outperforms the direct method in time. However, the speed up drops dramatically as more zeros are added.

In the $N = 21$, the speed was reduced by half by adding one zero in each dimension, showing a increased time complexity cost for the increase in accuracy. In the $N = 21$, the speed up drops by a third on $n_{\text{pad}} = 2$ to 20.97. However, zero-padding gives a degree of flexibility to the method to balance time and accuracy.

6.5 The Model Kinetic Equations and the Rel-ES Method

6.5.1 The BGK Model

One of the difficulties with dealing with the Boltzmann equation is that it depends on the product of two distribution functions [11], where

$$I[f] = \int (f_1' f' - f_1 f) g\, b\, db\, d\varepsilon\, d\vec{v}\,. \tag{6.1}$$

We drop the integral limits for simplicity. We aim to simplify the structure of the collision term while maintaining the same basic properties. One such method was formulated by Bhatnagar, Gross, and Krook which is known as the BGK model [4]. From Strutchup [15], we start by assuming that the post collision terms are close to Maxwellian, that is $f_1' \approx f_{M_1}'$ and $f' \approx f_M'$. We rewrite (6.5.1) as

$$\int f_{M_1}' f_M' - f f_1 g\, b\, db\, d\varepsilon\, d\vec{v}\,. \tag{6.2}$$

$\ln f_M$ is a linear combination of the collision invariants, so we have that $f_{M_1}' f_M' = f_{M_1} f_M$. We can then rewrite (**??**) as

$$f_M \int f_{M_1} g\, b\, db\, d\varepsilon\, d\vec{v} - f \int f_1 g\, b\, db\, d\varepsilon\, d\vec{v}\,. \tag{6.3}$$

Lastly, we assume that the difference between the two integrals may be neglected. This leads to the BGK collision term

$$\Psi_{\text{bgk}} = -\nu(f - f_M) \tag{6.4}$$

where

$$\nu = \int f_1 g \, b \, db \, d\varepsilon \, d\vec{v} \, . \tag{6.5}$$

We refer to (6.4) as the BGK collision operator. If ν is known, then the BGK collision operator can be computed fast. However, since ν is not always known during execution time, it is sometimes easier to calculate ν as

$$\nu = \frac{P}{\mu} \tag{6.6}$$

where P is pressure and μ is the dynamic viscosity.

An important property of the BGK collision integral is that it conserves the first three moments, that is

$$\begin{aligned} \int_{\mathbb{R}^3} \Psi_{\text{bgk}}(t, \vec{x}, \vec{v}) d\vec{v} &= 0 \, , \\ \int_{\mathbb{R}^3} v_i \Psi_{\text{bgk}}(t, \vec{x}, \vec{v}) d\vec{v} &= 0 \, , \quad i = 1, 2, 3 \\ \int_{\mathbb{R}^3} C^2 \Psi_{\text{bgk}}(t, \vec{x}, \vec{v}) d\vec{v} &= 0 \, . \end{aligned} \tag{6.7}$$

6.5.2 The ES-BGK Model

Calculation of the Prandtl number from the BGK model leads to a Prandtl number of 1 [15]. In order to recitfy this, Holway [9] proposed the ellipsoidal statistical BGK model, known as the ES-BGK model. In the ES-BGK model, the Maxwellian is replaced by an anisotropic Gaussian.

First define $\vec{c} = \vec{v} - \bar{\vec{v}}$. Next, we define the stress tensor, Θ, a 3×3 matrix such that

$$\Theta_{ij} = \frac{1}{n} \int_{\mathbb{R}^3} c_i c_j f d\vec{v} \,. \tag{6.8}$$

We also define $\mathbb{T} = (1-\alpha)RTI - \alpha\Theta$ where T is the temperature and I is the 3×3 identity matrix. α is an adjustment factor used to adjust the Prandtl number of the equation,

$$Pr = \frac{1}{1-\alpha} \,. \tag{6.9}$$

α must also be chosen such that $\mathbb{T}$ is symmetric positive definite (SPD). Zheng [16] showed that $\alpha \in [-1/2, 1)$ was sufficient to ensure SPD. We define the ESBGK collision operator as

$$\Psi_{\text{ES}} = -\bar{\nu}(f - f_{\text{ES}}) \tag{6.10}$$

where

$$f_{\text{ES}}(\vec{v}) = \frac{n}{(\pi^3 \det(2\mathbb{T}))^{1/2}} \exp(-\frac{\vec{c}^T \mathbb{T}^{-1} \vec{c}}{2}) \,. \tag{6.11}$$

Next, we prove an important property of the ES-BGK for the Rel-ES model.

Theorem 3 *Let $\mathbb{T}$ be SPD, then we have*

$$\frac{1}{n} \int_{\mathbb{R}^3} \vec{c}\vec{c}^T f_{ES} \, d\vec{c} = \frac{1}{n} \int_{\mathbb{R}^3} \vec{c}\vec{c}^T \frac{n}{(\pi^3 \det(2\mathbb{T}))^{1/2}} \exp(-\frac{\vec{c}^T \mathbb{T}^{-1} \vec{c}}{2}) = \mathbb{T} \,. \tag{6.12}$$

Proof. We ignore the constants for now and drop them from the integral, so we are left with

$$\mathcal{J} = \int_{\mathbb{R}^3} \vec{c}\vec{c}^T \exp(-\frac{\vec{c}^T \mathbb{T}^{-1} \vec{c}}{2}) \,. \tag{6.13}$$

Since $\mathbb{T}$ is SPD, we have that $\mathbb{T} = Q^T \Lambda Q$ for some orthogonal matrix Q and diagonal matrix Λ. We perform change of variables $\vec{w} = Q\vec{c}$ on (6.13). Since Q is orthogonal, we have that $d\vec{w} = \det Q \, d\vec{c} = d\vec{c}$. Multiplying (6.13) by Q on the left and Q^T on the

right, (6.13) becomes

$$\int_{\mathbb{R}^3} \vec{w}\,\vec{w}^T \exp(-\frac{\vec{w}^T\Lambda^{-1}\vec{w}}{2})d\vec{w}\,. \tag{6.14}$$

We can see that (6.14) is 9 equations. Focusing on one equation in (6.14), we get

$$\int_{\mathbb{R}^3} w_i w_j \exp((-\lambda_1^{-1}w_1^2 - \lambda_2^{-1}w_2^2 - \lambda_3^{-1}w_3^2)/2)dw_1dw_2dw_3, \quad i,j = 1,2,3. \tag{6.15}$$

When $i \neq j$ we get that the above integral is zero since the function is an odd function centered around zero. Without loss of generality, fix $i = 1$, $j = 2$. We get

$$\int_{-\infty}^{\infty} w_1 e^{-\lambda_1^{-1}w_1^2/2}dw_1 \int_{-\infty}^{\infty} w_2 e^{-\lambda_2^{-1}w_2^2/2}dw_2 \int_{-\infty}^{\infty} e^{-\lambda_3^{-1}w_3^2/2}dw_3.$$

We see that $\int_{-\infty}^{\infty} w_1 e^{-\lambda_i w_1^2/2}dw_1 = 0$ since it is an odd integral.

If $i = j$ (again, without loss of generality, set $i = j = 1$) in (6.15), then we have that

$$\begin{aligned}
&\int_{-\infty}^{\infty} w_1^2 e^{-\lambda_1^{-1}w_1^2}dw_1 \int_{-\infty}^{\infty} e^{-\lambda_2^{-1}w_2^2}dw_2 \int_{-\infty}^{\infty} e^{-\lambda_3^{-1}w_3^2}dw_3 \\
=&\left(\frac{2\pi}{\lambda_1^{-3}}\right)^{1/2}\left(\frac{2\pi}{\lambda_2^{-1}}\right)^{1/2}\left(\frac{2\pi}{\lambda_3^{-1}}\right)^{1/2} \\
=&(2\pi)^{3/2}\sqrt{\lambda_1\lambda_2\lambda_3}\lambda_1 \\
=&(2\pi)^{3/2}(\det\mathbb{T})^{1/2}\lambda_1\,,
\end{aligned} \tag{6.16}$$

where we used that $\lambda_1\lambda_2\lambda_3 = \det\Lambda = \det T$. We now have that (6.14) can be written as

$$(2\pi)^{3/2}(\det\mathbb{T})^{1/2}\Lambda\,. \tag{6.17}$$

Multiplying on the left by Q^T on the left and Q on the right to recover (6.13), we get

$$\begin{aligned}\mathcal{J} =&(2\pi)^{3/2}(\det \mathbb{T})^{1/2}Q^T\Lambda Q\\ =&(2\pi)^{3/2}(\det \mathbb{T})^{1/2}\mathbb{T}\,.\end{aligned} \tag{6.18}$$

We can now reintroduce the constants into (6.13),

$$\begin{aligned}\frac{1}{n}\frac{n}{(\pi^3\det(2\mathbb{T}))^{1/2}}\mathcal{J} &= \frac{1}{(\pi^3\det(2\mathbb{T}))^{1/2}}\mathcal{J}\\ &= \frac{1}{(\pi^3\det(2\mathbb{T}))^{1/2}}(2\pi)^{3/2}(\det \mathbb{T})^{1/2}\mathbb{T}\\ &= \mathbb{T}\end{aligned}$$

■

6.5.3 Rel-ES

6.5.4 Experimental Results: 0d Homogeneous Relaxation

In this section we present results of solution of the problem of spatially homogeneous relaxation using Fourier evaluation of the collision operator. Two cases of initial data were considered. In both cases, the initial data is a sum of two Maxwellian densities. In the first case, the dimensionless densities, bulk velocities, and temperatures of the Maxwellians are $n_1 = 1.0007$, $n_2 = 2.9992$, $\bar{\vec{u}}_1 = (1.2247, 0, 0)$, $\bar{\vec{u}}_2 = (0.4082, 0, 0)$, $T_1 = 0.2$, $T_2 = 0.7333$. These parameters correspond to upstream and downstream conditions of the Mach 3 normal shock wave. In the second case, we use the parameters of the example of the previous section: $n_1 = 1.6094$, $n_2 = 2.8628$, $\bar{\vec{u}}_1 = (0.7750, 0, 0)$, $\bar{\vec{u}}_2 = (0.4357, 0, 0)$, $T_1 = 0.3$, and $T_2 = 0.464$. These parameters correspond to upstream and downstream conditions of a Mach 1.55 shock wave.

For evaluation of the shockwave, parameters $M = 15$, $s = 5$ were used to evaluate the shockwave.

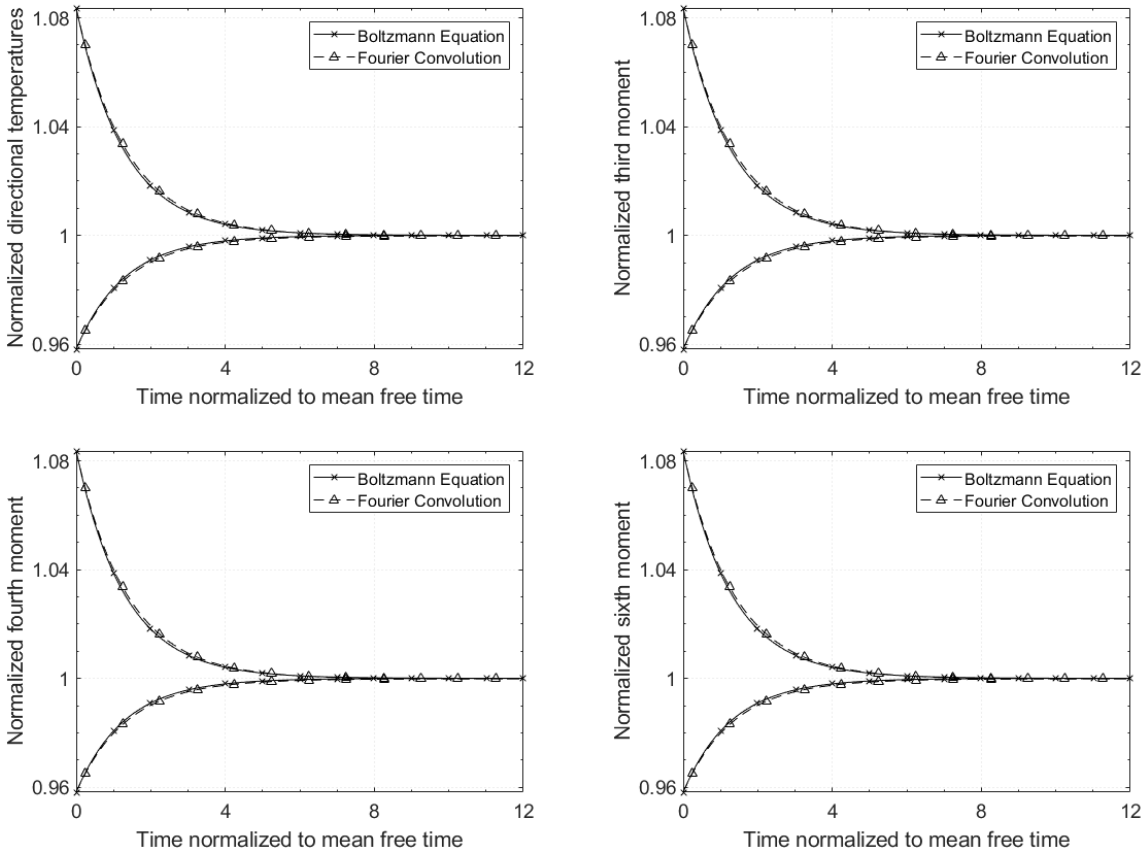

Figure 6.4: Relaxation of moments $f_{\varphi_{i,p}}$, $i = 1, 2$, $p = 2, 3, 4, 6$ in a mix of Maxwellian streams corresponding to a shock wave with Mach number 1.55 obtained by solving the Boltzmann equation using Fourier and direct evaluations of the collision integral.

References

[1] A. Alekseenko and E. Josyula. Deterministic solution of the Boltzmann equation using a discontinuous Galerkin velocity discretization. In *28th International Symposium on Rarefied Gas Dynamics, 9-13 July 2012, Zaragoza, Spain*, AIP Conference Proceedings, page 8. American Institute of Physics, 2012.

[2] A. Alekseenko and E. Josyula. Deterministic solution of the spatially homogeneous boltzmann equation using discontinuous galerkin discretizations in the velocity space. *Journal of Computational Physics*, 272(0):170 – 188, 2014.

[3] A. Alekseenko, T. Nguyen, and A. Wood. A deterministic-stochastic method for computing the boltzmann collision integral in $\mathcal{O}(mn)$ operations. *to appear in Kinetic and Related Models*, page 30p, 2015.

[4] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, May 1954.

[5] L. Boltzmann. Weitere studien u ber das warmegleichwicht unt gasmolekulen. 1872.

[6] Iain D Boyd. Vectorization of a monte carlo simulation scheme for nonequilibrium gas dynamics. *Journal of Computational Physics*, 96(2):411 – 427, 1991.

[7] I.M. Gamba and C. Zhang. A conservative discontinuous galerkin scheme with $o(n^2)$ operations in computing boltzmann collision weight matrix. In *29th International Symposium on Rarefied Gas Dynamics, July 2014, China*, AIP Conference Proceedings, page 8. American Institute of Physics, 2014.

[8] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer, 2007.

[9] L. H. Holway. New statistical models for kinetic theory: methods of construction. *Phys. Fluids*, 9(9):1658–1673, 1966.

[10] M.N. Kogan. *Rarefied Gas Dynamics*. Plenum Press, New York, USA, 1969.

[11] G. M. Kremer. *An Introduction to the Boltzmann Equation and Transport Processes in Gases*. Interaction of Mechanics and Mathematics. Springer, 2010.

[12] Armando Majorana. A numerical model of the boltzmann equation related to the discontinuous Galerkin method. *Kinetic and Related Models*, 4(1):139 – 151, March 2011.

[13] H. J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms*. Springer Series in Information Sciences. Springer-Verlag, Heidelberg, 1982.

[14] Lorenzo Pareschi and Giovanni Russo. Numerical solution of the boltzmann equation i: Spectrally accurate approximation of the collision operator. *SIAM Journal on Numerical Analysis*, 37(4):1217–1245, 2000.

[15] H. Struchtrup. *Macroscopic Transport Equations for Rarefied Gas Flows Approximation Methods in Kinetic Theory*. Interaction of Mechanics and Mathematics Series. Springer, Heidelberg, 2005.

[16] Y. Zheng and H. Struchtrup. Burnett equations for the ellipsoidal statistical bgk model. *Continuum Mechanics and Thermodynamics*, 16(1):97–108, February 2004.